EM511530759US

PATENT

Docket No. 277301

Box Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

NEW APPLICATION TRANSMITTAL

Transmitted herewith for filing is the patent application of

Inventor: PETROCY, RICHARD J.

For: SELF-ADDRESSING CONTROL UNITS AND MODULAR SIGN INCLUDING PLURALITY OF SELF-ADDRESSING CONTROL UNITS

1. Type of Application

This new application is for a(n):

X Original
__ Design
__ Plant

__ Divisional
__ Continuation
__ Continuation-in-part (CIP)

2. Benefit of Prior U.S. Application(s) (35 USC 120)

__ The new application being transmitted claims the benefit of prior U.S. application(s) and enclosed are added pages for new application transmittal where benefit of prior U.S. application(s) claimed.

3. Papers Enclosed Which Are Required for Filing Date Under 37 CFR 1.53(b) (Regular) or 37 CFR 1.153 (Design) Application

22 Pages of specification
3 Pages of claims
2 Pages of Abstract
20 Sheets of drawing
__ formal
X informal

4. Additional papers enclosed

__ Preliminary Amendment
__ Information Disclosure Statement
__ Form PTO-1449

__ Citations
__ Declaration of Biological Deposit
__ Submission of "Sequence Listing"
__ Authorization of Attorney(s) to Accept and Follow Instructions from Representative
__ Special Comments
__ Other

5. Declaration or Oath

__ Enclosed
executed by
___ inventor(s).
___ legal representative of inventor(s). 37 CFR 1.42 or 1.43
___ joint inventor or person showing a proprietary interest on behalf of inventor who refused to sign or cannot be reached.
_X_ Not Enclosed.

6. Inventorship Statement

The inventorship for all the claims in this application are:

_X_ The same

or

___ Are not the same. An explanation, including the ownership of the various claims at the time the last claimed invention was made,
___ is submitted.
___ will be submitted.

7. Language

_X_ English
___ non-English
___ the attached translation is a verified translation. 37 CFR 1.52(d).

8. Assignment

___ An assignment of the invention to ________________
____
___ is attached.
___ will follow.

9. Certified Copy

Certified copy(ies) of application(s)

____

____

from which priority is claimed

___ is(are) attached. A separate "Assignment cover letter accompanying new patent application" is also attached.

___ will follow.

10. Fee Calculation (37 CFR 1.16)

A. X Regular application

CLAIMS AS FILED

	Number filed		Number Extra		Rate	Basic Fee
						$770.00
Total Claims 37 CFR 1.16(c)	18	-20=	0	X	$22.00	
Independent Claims (37 CFR 1.16(b))	3	- 3=	0	X	$80.00	$
Multiple dependent claim(s) if any (37 CFR 1.16(d))			0	X	$200.00	$

___ Amendment canceling extra claims enclosed.

___ Amendment deleting multiple-dependencies enclosed.

___ Fee for extra claims is not being paid at this time.

Filing Fee Calculation $ 770.00

B. ___ Design application

Filing Fee Calculation $________

C. ___ Plant application

Filing Fee Calculation $________

11. Small Entity Statement(s)

__ Verified Statement(s) that this is a filing by a small entity under 37 CFR 1.9 and 1.27 is(are) attached.

Filing Fee Calculation (50%) $ 385.00

12. Request for International-Type Search (37 CFR 1.104(d))

___ Please prepare an international-type search report for this application at the time when national examination on the merits takes place.

13. Fee Payment Being Made At This Time

_X_ Not Enclosed

_X_ No filing fee is to be paid at this time.

___ Enclosed

___ basic filing fee	$________
___ recording assignment	$________
___ petition fee for filing by other than all the inventors or person on behalf of the inventor where inventor refused to sign or cannot be reached.	$________
___ for processing an application with a specification in a non-English language.	$________
___ processing and retention fee	$________
___ fee for international-type search report	$________
Total enclosed fees	$________

14. Method of Payment of Fees

___ Check in the amount of $_______.

15. Instructions as to Overpayment

___ Refund.

Signature of Attorney

Reg. No. 33,884

Tel. No. (201) 842-0800

Michael R. Friscia
FRISCIA & NUSSBAUM
405 Murray Hill Parkway
East Rutherford, NJ 07073
TEL: (201) 842-0800
FAX: (201) 842-0229

X Incorporation By Reference of Added Pages

X Plus added pages for New Application Transmittal where benefit of prior U.S. Application claimed

Number of pages added _1___.

___ Plus "Assignment Cover Letter Accompanying New Application"

Number of pages added ________.

__ Statement Where No Further Pages Added

__ This transmittal ends with this page.

PATENT

Attorney's Docket No. 277301

ADDED PAGES FOR APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED

16. Relate Back - 35 U.S.C. 120

X Amend the Specification by inserting before the first line the sentence:
"This is a

___ continuation
X continuation-in-part
___ divisional

of copending provision application(s)

X serial number 60/012,565 filed on February 29, 1996
X serial number 60/012,545 filed on February 29, 1996
X serial number 60/012,541 filed on February 29, 1996

17. Further Inventorship Statement Where Benefit of Prior Application(s) Claimed

a. This application discloses and claims only subject matter disclosed in the prior application whose particulars are set out above and the inventor(s) in this application are:

X the same
___ less than those named in the prior application

and it is requested that the following inventor(s) identified for the prior application be deleted:____________________

b. This application discloses and claims additional disclosure by amendment and a new declaration or oath is being filed. With respect to the prior application the inventor(s) in this application are:

___ the same
___ the following additional inventor(s) have been added:____________________

c. The inventorship for all the claims in this application are:

X the same
___ not the same, and an explanation, including the ownership of the various claims at the time the last claimed invention was made
___ is submitted
___ will be submitted.

18. Notification in Parent Application of this Filing

__ A notification of the filing of this

___ continuation
___ continuation-in-part
___ divisional

is being filed in the parent application from which this application claims priority under 35 USC 120.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner of Patents and Trademarks
Washington, D.C. 20231

Re: Our file: 277301
Applicants: Richard J. Petrocy
Serial No.:
Filing Date:
Title: SELF-ADDRESSING CONTROL UNITS AND MODULAR SIGN INCLUDING PLURALITY OF SELF-ADDRESSING CONTROL UNITS

Office:
Examiner:

Sir:

Enclosed for filing in the United States Patent and Trademark Office is the following:

1. New Appln. Transmittal
2. Patent Appln. (Pages 27)
3. Appendices A, B, C
3. Drawings (Pages 20)
4. Postcard

CONDITIONAL PETITION

If any extension of time is required for the submission of the above-identified items, Applicant requests that this be considered a petition therefor.

Respectfully submitted,

Michael R. Friscia
Registration No. 33,884
Friscia & Nussbaum
405 Murray Hill Parkway
East Rutherford, NJ 07073

2-28-97
Date

enc.

I hereby certify that this correspondence is being deposited with the United States Postal Service, postage prepaid, as "Express Mail Post Office to Addressee," Mailing Label No. EM511530759 to The Commissioner of Patents and Trademarks, Washington, D.C. 20231 on 2-28-97.

By:

EXPTRANS.277

20

EM511530759
A/Nv
Fee

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTORS: PETROCY, RICHARD J.

TITLE: SELF-ADDRESSING CONTROL UNITS AND MODULAR SIGN INCLUDING PLURALITY OF SELF-ADDRESSING CONTROL UNITS

SPECIFICATION

BACKGROUND OF THE INVENTION

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Applications Serial Nos. 60/012,565, 60/012,545, and 60/012,541 filed February 29, 1996. The entire disclosures of the applications are expressly incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates to a protocol for self-addressing control units, and more particularly to a modular sign comprising a plurality of self-addressing control units positioned side by side to form an array, each of the control units having a mechanical sign mechanism for displaying one of a plurality of characters to display a message on the array, which sign can be controlled from a remote location to change the characters displayed by the control units to create and change messages on the sign. Additionally, the present invention relates to the use of a protocol for self-addressing control units for application in any field wherein a plurality of control units are used in a system. Additionally, the present invention relates to a method and apparatus for installing a plurality of control units to form an array.

RELATED ART

In the past signs have been made to have a single image thereon for the life of the sign. Of course, the entire face of the sign cold be replaced with a new face. Additionally, it is known to provide signs that can be backlit and have, on the face thereof, slots for holding individual clear panels with characters thereon so that such characters can be arranged to form words. This type of sign is used on movie theater marquees to display the names of the movies playing at the theater, and the times that would such movies are scheduled to begin. However, this type of sign is difficult to install. Additionally, in order to change the names on the sign, one needs to either lower the sign down to ground level or use a ladder to climb up to the sign and remove the panels from the sign and put on new panels bearing the proper characters to spell the proper word to indicate the name of a new movie. Besides being dangerous, this procedure is time intensive. Additionally, this process must be performed frequently, such as on a weekly basis, which compounds the amount of time involved. There is additional time involved in replacing fluorescent bulbs which provide the back lighting for the sign as they burn out.

A prior attempt at overcoming these problems is found in Lesko, et al., U.S. Patent No. 5,061,921 disclosed a remote-controlled message sign which is controlled by a pager which receives radio signals from a paging service and provides output signals in response to the radio signals. The output signals of the pager are used to control one or more drive motors which move a multiple position message device to a desired position. The display device includes a wheel or drum having an outer cylindrical surface and an axle and is rotatably mounted on the sign. A motor drive rotates the wheel to position the desired letter or number in the window of the sign. A position data reader on the drum determines the position of the drum relative to the window of the sign. However, this does not overcome all of the problems in the prior art.

Another attempt at providing an automatically changeable display sign is disclosed in Daugherty, et al., U.S. Patent No. 5,184,116 for a back-lightable diffusive sign for displaying alphanumeric characters and graphics comprising a plurality of mechanically moveable elements, each have a dark translucent face and a bright translucent face which are moveable from one to the other face interchangeably by a series of electromechanical driving elements. However, this sign does not overcome all the problems associated with the signs of the prior art.

Accordingly, what is desired, but has not heretofore been achieved, is a sign for displaying messages which messages can be inexpensively and easily changed from a remote location.

Additionally, it has been known is the past to provide a series of control units, such as computers, computer networks, or other controllers, for performing a desired function. In the past, efforts at coordinating the outputs of the various control units involved wiring each separate control unit directly to a main controller to form an electrical and mechanical link. Such a method however, is expensive based on the wiring involved. Another method of linking the control units together is by means of multiplexing which involves an array of many "X" and "Y" connecting wires from the main controller to each of the control units. Further, it is known to serially or sequentially link a main controller to control units by having the installer set switches on each of the control units. Indeed, many of the networking cards currently in use in computer networks are configured by the manufacturer to have a certain switch sequence for identification purposes, and these control units are mixed and matched, but the problem sometimes arises that more than one control unit has the same identification number and causes confusion in the network. All of these methods are material intensive in terms of wires and/or labor intensive and/or require expert installers to understand and install each system and/or are limited by the manufacturer of the units.

Accordingly, what is desired, and has not heretofore been invented is a control unit capable of using one single data path (one wire or parallel wires or fiber optic or radio path) where all of the units are addressed sequentially and set their own addresses based on the referencing of the prior unit to self-address and to self-install without the aid of a technician.

Additionally, in the past there has been a problem with hanging signs and running electricity thereinto. Signs had to be separately, mechanically, installed and separately, electrically, interconnected. For modular signs there has been a problem installing a plurality of units need at an even and aligned position. It is difficult to achieve such alignment because of the measuring that must take place to insure that units are mounted at a aligned height with proper spacing therebetween.

Accordingly, what is needed, and has not heretofore been available, is a method for mounting and electrically connecting a plurality of units which compensates for improper installation.

OBJECTS AND SUMMARY OF THE INVENTION

It is a primary object of the present invention to provide a protocol for self-addressing contol units.

It is an additional object of the present invention to provide self-addressing control units which periodically re-address themselves.

It is a further object of the present invention to provide self-addressing control units which look at the previous control unit identification, add a one thereto, and store the result as the address of the control unit.

It is still a further object of the invention to provide a plurality of self-addressing control units which do not require dip switches or custom program chips for addressing.

It is an additional object of the present invention to provide control units which do not have to be set up by a skilled electrician or a computer installer.

It is an additional object of the present invention to provide an array of self-addressing control units wherein if one control unit is damaged, the remaining control units can continue to operate separately and independently.

It is another primary object of the present invention to provide a modular sign comprising a plurality of self-addressing control units, wherein each of the control units can display a character to form a message on the modular sign.

It is an additional object of the present invention to provide a sign comprising a plurality of self-addressing control units to provide a message which message can be remotely controlled and remotely changed.

It is an additional object of the present invention to provide a modular sign having a plurality of control units which may be controlled by a telephone modem interface.

It is an additional object of present invention to provide a modular sign having a plurality of self-addressing control units which may be controlled by a pager interface.

It is another primary object of the present invention to provide a method and apparatus for installing a plurality of control units to form an array.

It is another object of the present invention to provide an installation apparatus which includes mechanical attachment means and electrical communication means integrated into one unit.

It is an additional object of the present invention to provide a method and apparatus for installation of a plurality of control units to form a modular sign which does not require a wire harness.

It is an addtional object of the present invention to provide a method and apparatus for installing an array of control units to form a modular sign which includes a "reverse" bus system.

It is an additional object of the present invention to provide an array of control units having a reverse bus system, wherein the bus is formed on circuit boards within the control units, and the control units are interconnected other adjacent control units by electrical contacts within the installation brackets.

It is an additional object of the present invention to provide a method and apparatus for installing an array of self-addressing control modules which can be installed by one who is not skilled in the sign installation business and one who is not a skilled electrician.

These and other objects are achieved by the protocol for self-addressing control units of the present invention. The protocol is effected by arranging a plurality of control units in a sequence and running a line from a master controller with links off the line to each control unit. Additionally, a feedback line is provided in the reverse direction for each control unit to communicate backwards with the previous unit. The master controller sends out a signal to identify itself as 00 and the control units down the line address and identify themselves by adding a 1 to the number that it sees. Accordingly, the first control unit addresses itself as 1, the second control unit addresses itself as 2, etc. This protocol can be implemented on a row by row basis, or in one line extending through a plurality of rows. This protocol has applicability to modular signs as well as other fields of application of wherein a number of control units are linked together such as a computer networking, prosthetics, etc.

When used in connection with a modular sign, the protocol of the present invention can be used to coordinate displaying a message by allowing each of a plurality of control units to display a desired character to form a message on the array of control units. This sign can be remotely controlled by a pager system. Each control unit includes a box housing a Mylar scroll operated by a motor and employing an optical sensor to read markings on the Mylar scroll to position appropriate characters in response to a signal to display a character to form a part of a message on the modular sign. The box includes an open face with a frame therearound which is a black opaque color. A transparent cover sits thereover to seal up the control unit. The control units are positioned side by side to form an array. The control units can be removed and serviced and/or replaced by means of extraction tools.

The control units are mounted against a wall or within an enclosure by means of connecting brackets having attachment means on upper and lower ends thereof, and include a plurality of contacts formed within receptacles positioned along the brackets to receive spades extending from the back of the control units. Accordingly, the mounting brackets provide electrical contacts as well as mechanical attachment for the control units. The control units include circuit boards in communication with the spades having a reverse bus formed on the circuit board to run power and data along the system, the mounting brackets serving to provide electrical communication between the units and to support the units in an array.

BRIEF DESCRIPTION OF THE DRAWINGS

Other important objects and features of the invention will be apparent from the following Detailed Description of the Invention taken in connection with the accompanying drawings in which:

FIGS. 1a and **1b** are front plan views of a modular sign of the present invention.

FIG. 2 is a perspective view of a single control unit or module of the present invention.

FIG. 3 is a top view of a plurality of control units arranged together to form an array for displaying a message in the form of a sign, and also shows extraction tools for removing control units from the array.

FIG. 4 is a circuit diagram of a parallel shift register which can be used address control units in the present invention.

FIG. 5 is another embodiment of a circuit for addressing control units of the present invention.

FIG. 6 is another embodiment of a circuit for addressing control units of the present invention.

FIG. 7 is an alternative view of a system shown in the circuit diagram of **FIG. 6**.

FIG. 8 is a chip input/output configuration for a chip used for the system shown in **FIGS. 6** and 7.

FIG. 9 is an actual working schematic circuit diagram of the circuit for use and connection with the system of **FIG. 5**.

FIG. 10 is an actual working schematic circuit diagram of the circuit for use in connection with the system of **FIG. 4**.

FIG. 11 is a bus diagram for use in connection with the reverse bus system for the system of **FIGS. 6** and 7 of the present invention.

FIG. 12 is a block diagram of the theory of operation showing the circuit boards of a plurality of controllers and control units interconnected together.

FIG. 13a, **b**, and **c** show a logic flow chart of the system of **FIGS. 6** and 7.

FIG. 14 shows a circuit diagram for the circuit boards for the system shown in **FIG. 6**.

FIG. 15 is a perspective view of the connector used to mount the control units of the present invention.

FIG. 16a shows a side view of the arrangement of the electrical and mechanical contacts within the channels formed within the period.

FIGS. 16b and **16c** show other embodiments for the shape of the contacts.

FIGS. 17, **18**, **19** and **20** show the connectors attached to a wall to position the connectors at relatively uneven positions along a wall while maintaining even positioning of control units attached thereto.

FIG. 21 is a schematic of a circuit for use with the circuit board of **FIG. 14**.

FIG. 22 is parts list of the components of the circuit diagram shown in **FIG. 21**.

DETAILED DESCRIPTION OF THE INVENTION

Referring to **FIGS. 1a** and **1b**, a front plan view of the sign **10** of the present invention is shown with a first message in **FIG. 1a** and a different changed message in **FIG. 1b**. The sign includes a plurality of control units or modules **20** arranged along side each other to form the sign. Each control unit or module **20** is capable of displaying a desired character such as a number or letter so that the sign **10** can display a desired message. The control units can also be placed sideways as shown.

Referring to **FIG. 2**, each control unit **20** comprises a box-like enclosure **22** with a cover **24** that fits thereon and snaps thereover. To retain the cover **24** in place on the box enclosure **22**, a cooperating protrusion formed on the box **22** can coact with a recess formed within the side wall **26** of the cover **24** to retain the cover **24** on the box **22**. Preferably, the box **22** is a black opaque color and the face **28** of the cover **24** is transparent. The box **22** preferably includes a front frame **23** which is also an opaque black color to frame out the display area therewithin which is covered by the cover **24**. By forming the frame **23** on the box **22**, the advantage of a uniform color match is obtained which may not be obtained if the frame was painted on to the cover **24**. Additionally, the frame eliminates the cost associated with painting the cover, and this construction allows the box **22** to be formed of a different material from the cover **24**.

The construction of the control unit **20** allows for large surface signs formed from a plurality of units **22** to be flat, water-tight, able to expand and contract over irregular surfaces, and still be pleasing to the eye. Additionally, the overall affect of a plurality of control units **20** grouped together forms a sign of an aesthetically pleasing appearance without the need for fasteners and seems required with conventional sign faces. Additionally, this modular sign is vandal-proof because there are no exposed fasteners or edges to grip without the aid of an extraction tool.

Referring to **FIG. 3**, a sign **10** is shown having a plurality of modules **20** each of which are arranged along side each other to form a modular sign. Each control unit includes a box **22** and a cover **24**. The boxes **22** are mounted in a side by side relationship by mounting means which will be hereinafter discussed. Once installed, a module **20** cannot be easily removed as there is no area to grab on to the box **20**.

Extraction tools **30** may be used to extract a module **20** from a sign **10**. The extraction tools comprise a grip means **32**, an insertion portion **34**, and an engagement portion **36** which is bent back against the insertion portion **34** to form a small angle between the insertion portion **34** and the engagement portion **36**, which ends in a point **35**. Accordingly, in order to extract a module **20** from a sign **10**, two extraction tools **30** are inserted along the sides the module **20** to be removed by gripping the insertion tool **30** by the grip means **32**, inserting the insertion portion **34** and the engagement portion **36** along sides the module **20** to be removed to insert the point **35** and the engagement portion **36** past a lip formed by the side wall **26** of the cover **24**. Once the engagement portion **36** bypasses the side wall **26** of the cover **24**, the engagement portion **36** is naturally biased to spring away from the insertion portion **36** to align with the lip formed by the side wall **26** of the cover **24**. The engagement portion **36** engages the lip of the side wall **26** of the cover **24** and then one can pull the extraction tool **30** by the grip means **32** to pull the module **20** away from the mounting means in the direction of arrow A to remove the module **20** from the sign **10**.

The construction of the modular sign **10** of the present invention permits a sign to be constructed that is serviceable from the front with no external cover plates which can buckle and which need to be seamed together. This allows retrofitting of existing boxes to make aesthetically pleasing signs of 30 feet or more in size with a commercially appealing look. Without the covers **24**, there would have to be secondary water tight cover plates with seals and gaskets to encompass the entire sign. Of course, such a construction is also within the scope of the present invention. Service panels would have to be provided on the rear of the sign making retrofitting of existing signs possible.

The present invention includes a method and apparatus for addressing and identifying the control units comprising a system based on a self-addressing protocol. This protocol can be implemented in a number of different ways. As shown in **FIG. 4**, a parallel wire bus with a BCD code using four wires in parallel and one wire as a clock pulse to set a four bit latch to trap the data sequentially can be used. This method is known as a paralleled shift register and is used to trap data in memory boards on computers. What is different in the present system is that many separate circuit boards, each one located in a separate control unit or module, runs different applications and the data must be shifted along the wires two bits at a time to allow each unit to trap its data.

FIG. 5 is a circuit diagram showing a system having a reduced amount of wires to send data and simplify the board latching design by using a micro controller to reduce the transmission lines to two lines. In this embodiment, data is sent by sequential shifting and the data is received through one or more trappings of data one bit at a time. This reduces the speed of this system, and because of the sequential nature of the system, if one unit goes down, the system cannot work.

FIG. 6 shows another embodiment of a circuit for addressing control units wherein two or three wires are used to control the units and the data flow to the units. In this system each of the control units self-addresses itself upon system startup. This is accomplished by each unit checking its ID number by looking at the ID number of the unit in front of it and adding a one to that number and storing that number in a permanent non-volatile memory establishing its ID. This happens down the line and accordingly, an infinite amount of sequential control units can self-identify themselves in the system.

When the unit knows its ID number it watches the main broadcast wire or fiber optic link or radio link or other communication means for its ID number. When it sees its ID number, it reads the block of data that follows it and traps that data. Accordingly, all of the units constantly look at the broadcast line to obtain data. If any of the control units should fail, the remainder of the units are able to function independent of the failed unit. Additionally, a failed unit can be replaced by any other operable unit, even one already in the system with another assigned number, and the replacement unit will appropriately address itself and will be active in the system. In this way a system of many control units or parallel computers is created, which units self-address and are able to look to a broadcast line to trap relevant data directed to each of the units, and the units can each perform a task as a collective unit. This system comprising a plurality of control units or parallel computers may be serviced by a person having no knowledge of the system by merely replacing failed units. The failed units then re-address themselves and function as part of the system. If that unit fails, the rest of the system still continues to function.

FIG. 7 is a alternative view of the system depicted in **FIG. 6**. As can be seen in **FIG. 7**, a key or master control units sends data along a wire. Meanwhile, the key sends out a signal to the first unit to address itself as unit 1. Thereafter, the second unit addresses itself as two by seeing the first and adding a one thereto. This is continued down the line so that each unit self-addresses itself. Further, it should be pointed out that the units can be addressed in a single sequence or each row can be separately addressed: Row 1 comprising Unit 1,1; 1,2; 1,3; etc., and Row 2 comprising Unit 2,1; 2,2;

2,3; etc.

FIG. 8 shows a diagram of a chip input/output configuration for a chip to be used with the system shown in **FIG. 6** and 7.

FIG. 9 is an actual working schematic circuit diagram for use in connection with the system shown in **FIG. 5**.

FIG. 10 shows an actual working schematic circuit diagram for use in connection with the system shown in **FIG. 4**.

FIG. 11 shows a bus configuration for the systems of **FIGS. 6** and 7.

FIG. 12 is a block diagram of theory of operation showing two rows having two columns of a circuit and chips for running the system shown in **FIG.** 7. Note that each row has a key having a computer chip, a beeper with RS-232 output and/or a phone line with RS-232 output interconnected with the computer chip as well as a power source interconnected with a computer chip and lines leading from the key along the column to contact a first control unit where the power supply is brought to the first unit and a line for the chip ID is interconnected with the control unit. Additionally, there is a link to the control unit for providing a feed back line and there is a link from a one controller for a first row to a second controller at a second row. Alternatively, there could be one controller controlling all of the columns and rows. Each control unit includes a computer chip which ties into the chip ID line coming from the key and that extends out to a subsequent chip ID which would again interconnect with a subsequent control unit. Additionally, the power source brought in from the controller is run in to the control unit and used to power the control chip and then is brought through the control unit to subsequent control units. The computer chip is further interconnected logically with a motor driver and a motor which mechanically interconnected with a

Mylar-type scroll mechanism having a plurality of characters thereon which can be moved to position a desired character at a desired location. Additionally, the computer chip is interconnected with a photo sensor for identifying a bar code or other identification means associated with the Mylar-typed scroll to properly position the desired character at a desired location by reading the bar code off the Mylar-type scroll. Finally, the computer chip is also interconnected into the feedback line to communicate with the prior control unit or ultimately the main controller. The subsequent control units are interconnected with previous control units in the same way and subsequent rows are interconnected with additional controllers or the main controller.

Each box includes a transformer to avoid custom switching supplies. In the key module, each one needs a power supply as big as it is because the motor draws the most amount of power, but for broadcasting the motor is not running the units steal power from the first module and do not need to have a power supply.

FIG. 13 shows a logic flow chart for a control of each box from power-on for system of **FIG. 6**. Initially, the system must go through a setup sequence. The first thing the computer needs to know is if it already knows its ID number. If it knows it, it jumps right down into "Do I know where I move to." If it doesn't know it, then it is going to look to its key module to center itself with the module back and forth, find the bar code, come back in there and look for its address and set the address at E^2 which is non-volatile memory, or Electronic Erasable Memory. Then it turns the left control on, then turns the motor on, then it reads the photo cell to see if there is black. If there is black, then it sets the time. If it is not black, then it turns the motor on to move it to a white position. Then it turns the motor back to the right and it says where I am. Anotherwords, it takes the Mylar and moves it to the very beginning of the row. It will see black, white, black, black, white. It looks for that real long black mark and then it creeps back to where the edge is and says OK. If it already knows where it is, then it does not move the module. Then it looks for the address. If it knows where it is at, then all it does is it looks to see whether or not compare where it is with the new data. The new data comes in an E^2 code in front of it. Then it is waits until it gets some new data in. When it gets the new data in, it takes the new data and moves the Mylar appropriately to get to the new spot. Once it sets the direction, then it turns the motor on because the direction is one wire and the on/off is another wire. It is going to look for the black, set the time, and look for black again. Now the reason why there are multiple blacks in here is because the first black if it looks for black it needs to see that black in for a certain amount of time because it could be a scratch and it is called debouncing. So it goes through a loop and looks further for black. If it sees black but then doesn't see black again, it thinks that the black was just a false black, it is not long enough to be a code, ignore it, and goes back for a loop. Once it finds the black, then it measures the black to see if the black is less than a certain amount, that tells that it is a small one. If it is longer than the amount, then that tells it is the long black mark because there is a long black mark and a short black mark. Once it checks that it does count a number and gets an address number, is the number short, is it out, check for odd and even and if it is bad, add one to the count and send it back. If it is good, ignore it, check the data and the count, latch the data, permanently store it, and then tell the computer in E^2 memory.

A copy of a computer program for running the circuitry of **FIG. 5** is attached hereto as appendix A.

Importantly, the protocol comprising a plurality of modules wherein each of the modules comprises a separate discreet mechanism which operates in unison with the other modules to create a system. Importantly, each of the modules is self-addressing is self-identifying and accordingly, the system has a high degree of surviveability and is easily maintained and fixed. The system of the present invention has applicability to modular signs as discussed herein as well as applicability to computer network systems wherein a plurality of computers are placed on a network and each computer has to be identified in order to properly communicate and interact with the main controller as well as with the other computers. Following the protocol of the present invention, each computer would self-address itself and accordingly, be replaceable with any of the other modules to continue to properly work. The protocol of the present invention has further utility in application to a prosthetic type device which involves a plurality of modules for communicating information and taking specific action.

For example, a prosthetic device comprising a hand, in a simplified form, could comprise six different modules, one for each finger, and one for the palm and one for wrist and one for arm. Each of these modules would be self-addressing and accordingly, the thumb could identify itself as number 1, the index finger is number 2, the middle finger is number 3, etc. Thereafter, each of the modules watches the information line for information relating to the particular module. For example, the index finger monitors the information bus for a signal identifying module 2. If such a signal comes through to module 2, module 2 then looks for the subsequent information or data which describes the action that it should take. Accordingly, for finger number 2 to move, it looks for its identification number and then for data which tells it to move and upon receiving that data it appropriately moves. In the system, should the hand or thumb fail, the index finger can still operate independently through software that allows it to still work in a limp mode albeit less efficiently because it sees all the data. Additionally, if the thumb is replaced, it addresses itself and becomes part of the system without the protocol thereto.

FIG. 14 shows a circuit board included in the control units of the present invention.

FIG. 15 is a perspective view of the connector **60** used to mount the control units of the present invention.

FIG. 16a shows a side view of the arrangement of the electrical and mechanical contacts within the channels formed within the period. Accordingly, the connector **60** includes the a base **62**, a plurality of upstanding wall **64** interconnected with the base and extending perpendicular therefrom, retainers **66** positioner at the upper end of the upstanding walls, and electrical contacts **68** positioned within the spaces formed by the upstanding walls and retained within the connector by means of retainer **66**. The contacts are preferably gold plated to resist corrosion. The contacts provide mechanical support for the control modules and additionally provide for electrical connection between adjacent control units. The connectors further include side walls **70** to form the connectors into a unit. Additionally, apertures **72** are positioned at upper and lower ends of each connector to facilitate connection of the connector to a wall or enclosure or other location for fixing the connectors thereto.

FIGS. 16b and **16c** show other embodiments for the shape of the connector. Also, it should be noted that the connectors and/or the knife contacts from the control units can have a protrusion to retain the interconnection between the knife contacts and the contact **68**.

Referring to **FIGS. 17**, **18**, **19** and **20**, it can be seen that the connectors are attached to a wall by means of inserting connectors through the apertures to position the connectors at relatively even positions along a substrate. Each control unit can then be mounted on adjacently positioned contact connectors. Each connector is large enough to receive the knife connectors of adjacent control units to provide electric connection therebetween. Additionally, it can be seen from **FIGS.17-20** that the positioning of adjacent connectors does not have to be perfect in order to provide for a uniform appearance of the control units attached thereto. Anotherwords, there is leeway between the positioning of the connectors and the overall appearance of the array of control units connected therewith. The receptacle in the connectors allow for the interconnection of knife contacts from the control modules to tie the control modules together electrically and mechanically. The connectors allow for multiple units to be fastened to a wall or board without any external wiring. The connectors allow high currencies while data passes through an entire array of control units providing the current data evenly to all units without the need for wiring. As much as two inches of latitude is provided allowing for improper installation of the connectors while still making a uniform array of modules to form a uniform looking sign. Further, the size of the connector allows for up to four degrees of canting due to improper installation or due to an irregular wall behind the connectors and allows the modules to still provide a uniform look to the array. The large size of the connector allows for the handling of high currence without over heating and maintains compliance with the National Electrical Code. Additionally, the large size allows for a physically sufficient mechanical connection to secure heavy mechanical objects as a final attachment point without the need for external fasteners. Further, the pertrusion or dimple that is provided on the male spade or on the contact itself prevents the walking of the control unit out from the connector due to vibrations.

With respect to prosthetics, the individual direct commands that have to go through the hierarchy, but on top of that are generalized global commands. So as well as the self-addressing routing, there are some global commands that all of the units look for which can supersede local commands through separte routines through separate key words. It takes more processing time but because all of the modules are listening to the data line, the path of communication broken.

FIG. 21 is a schematic of a circuit for use with the circuit board of **FIG. 14**. **FIG. 22** is parts list of the components of the circuit diagram shown in **FIG. 21**. Appendix B is a transmission code for the key module. Appendix C is the actual code for the module itself.

Having thus described the invention in detail, it is to be understood that the forgoing description is not intended to limit the spirit and scope thereof. What is desired to be protected by Letters Patent is set forth in the appended claims.

CLAIMS

What is claimed is:

1. A system for self-addressing one or more control units comprising:

controller means;

one or more control units;

electrical communication means extending between the controller means and the one or more control units;

signal means for requesting the control units to identify themselves;

means for each control unit to separately identify itself by receiving a number input from the a previous control unit and adding a one thereto.

2. The system of claim 1 wherein each control unit means includes a non-volatile memory in which it stores its identification number.

3. The system of claim 2 wherein each control unit has a feedback line to anohter control unit

4. A self-addressing control unit system comprising:

a plurality of control units electrically interconnected by a bus;

a control means electrically interconnected with the plurality of control units by the bus;

broadcast means associated with the controller means for broadcasting a signal to the control units along the bus;

means for each control unit to identify itself by receiving an identification number of a previous control unit, adding a one thereto, and storing that number in memory;

5. The apparatus of the claim 3 wherein the control units look to the broadcast wire for an ID number and read a block of data that follows its ID number.

6. A method for networking a plurality of control units comprising the steps of:

providing controller means;

electrically interconnecting a plurality of control units with the controller means;

sending a system start-up signal from the controller means to the plurality of control units;

sequentially self-addressing the plurality of control units by having a controlled unit look at the address of a pervious control unit of the plurality of control units, add a one to the address, and store the address in memory.

ABSTRACT

A protocol for self-addressing control units is effected by arranging a plurality of control units in a sequence and running a line from a master controller with links off the line to each control unit. A feedback line is provided in the reverse direction for each control unit to communicate backwards. The master controller sends out a signal to identify itself and the control units down the line address and identify themselves by adding a 1 to the number that each control unit receives from the previous control unit. Accordingly, the first control unit addresses itself as 1, the second control unit addresses itself as 2, etc. This protocol has applicability to modular signs as well as other fields of application of wherein a number of control units are linked together such as a computer networking, prosthetics, etc.

When used in connection with a modular sign, the protocol of the present invention can be used to coordinate displaying a message by allowing each of a plurality of control units to display a desired character to form a message on an array of control units. This sign can be remotely controlled by a pager system. Each control unit includes a box housing a Mylar scroll operated by a motor and an optical sensor to read markings on the Mylar scroll to position appropriate characters in response to a signal to display a character to form a part of a message on the modular sign. The box includes an open face with a frame there around which is a black opaque color. A transparent cover sits thereover to seal up the control unit. The control units can be removed and serviced and/or replaced by means of extraction tools. The control units are mounted against a wall or within on an enclosure by connecting brackets including a plurality of contacts formed within receptacles positioned along the brackets to receive spades extending from the back of the control units. Accordingly, the

mounting brackets provide electrical contacts as well as attachment for the control units. The control units include circuit boards in contact with the spades and a reverse bus is formed on the circuit board to run power and data along the system.

Seve vow

Appendix A (7 pages)

```
; START OF PROGRAM DATASIGN EXPERIMENTAL CODE
; FOR USE BY DATASIGN
; based on serdata4.src for use with arrow message pointer default=1 9/13/94
bit_K       =     24          ;Change this value for desired baudrate is 19.2KBaud for 8 Mhz,9600 Baud
for 4 Mhz
half_bit     =     bit_K/2        ;as shown in table.
;
TOP1      EQU  RA.0; TOP FIRST BIT
BOT       EQU  RA.1; BOTTOM BIT
optoset   equ  ra.3;set data pulse normally high
serial_in  equ  rb.1
direction  equ  rb.6; output
on_off    equ  rb.5; output
Data_clear =  rb.4; output change for pic1654j.pcb artwork
reset_in   =  rb.2
optopwr    =  rb.0
OPTO       EQU  RA;   REFERS TO ALL 4 PINS AS inputs ra.2 & ra.3 tied HI
BRAKE      EQU  rb.7; use motor chip BRAKE input for quicker stops
Shoneytape EQU  rb.3; shoney tape=1, else honey tape
;
                org   8
```

Set aside space for VARIABLES

Gen RAM → establish start address

```
delay_cntr     ds     1        ;Counter for serial delay routines
bit_cntr       ds     1        ;Number of received bits
rcv_byte       ds     1        ;The received byte
rcv_done   ds 1; its done
Count0     ds  1;Register labels
Count1     ds  1
Number     ds  1
RcvReg     ds  1
DlyCnt     ds  1
; datasign start
DEFAULT    ds 1
lastletter ds 1
newdata    ds 1
Count      ds 1
Datain     ds 1
optostop   ds 1
;
;Flags
FLAG       EQU  1AH.0
lastdirection  EQU  1AH.1
botFLAG    equ  1Ah.2
TOP1FLAG   EQU  1AH.3
FLAG2      EQU  1AH.4;
dataflag   equ  1Ah.5
jumpflag   equ  1Ah.6
R_DONE     equ  1AH.7
; datasign end
; Org 0 sets ROM origin to beginning for program.
```

9

10

"=" = EQU

ORG =

DS = VARIABLE SPACE

5/4

PeP

D 3/4"

GP Re[illegible]

24 NAME

```
                org     0
;
; Set the device type, oscillator type, watchdog timer status, and code
; protect status
;
            DEVICE  PIC16C54,XT_OSC,WDT_OFF,PROTECT_OFF
;
            RESET   Start ;Set reset vector to address at Start
                    ;(PIC will jump to this when reset)
;
Start       clrB Flag
            clrb flag2
;                   76543210 bit registers
            mov !RA,#00001111b ;Set data direction register for port A 4 INPUTS
            mov !RB,#00001110b;Set data direction register for port B 6/28/94, 1,2,3 input

            clrb lastdirection
            CLRB BOTFLAG
            CLRB TOP1FLAG
            CLRB R_DONE
            clrb dataflag
            setB BRAKE
; new code 9/12/94
            jb optoset,notarrow
            Mov optostop,#4; ra.2 tied HI all else 0
            MOV DEFAULT,#5;blank character Arrow tape
            jmp resume
notarrow    Mov optostop,#12; ra.2 & ra.3 tied HI all else 0
            jb Shoneytape,shoneyyes
            MOV DEFAULT,#23;blank character HONEY tape normally 22 see Joel
            jmp resume
shoneyyes   MOV DEFAULT,#17;blank character SHONEY tape normally 16 see Joel
; end new code
resume
;
            clrb optopwr; turn on FET for optoLEDs
;
            clrf Count
            clrf Count0
            clrf rcv_byte
            setb rcv_done
            clrb Data_clear
;

            mov lastletter,#60;
            MOV NUMBER,#0

            call end_delay; wait a bit
;
            clrB on_off; turns motor on
            setB direction; go to beginning { rewind }
```

```
          cje opto,optostop,failsafe;6/29/94 change VK 9/13/94
;initialize loop
INITIALIZE JNB FLAG2,BOTLOOP; debounce routine for opto tops and bottom
          CSE OPTO, optostop;both tops and bottom goto 0 at start of tape
          JMP INITIALIZE
;
failsafe  clrb direction; got to start now go forward
Digit     JNB FLAG,UPLOOP
;
          JB TOP1,DIGIT
;
FORWARD    INC NUMBER
          cjne Number,default,Clear
          mov NEWDATA,Number ;save Number
          setb flag2 ;
          jmp center
Clear     clrb flag
          jmp Digit
;
; **********************************************
;
; delays
; This delay loop takes four instruction cycles per loop, plus eight
; instruction cycles for other operations (call, mov, the final djnz, and ret).
; These extra cycles become significant at higher baud rates. The values for
; bit_K in the table take the time required for additional instructions into
; account.
bit_delay      mov     delay_cntr,#bit_K
:loop          nop
                  djnz    delay_cntr, :loop
                  ret
;
; This delay loop is identical to bit_delay above, but provides half the delay
; time.
;
start_delay     mov     delay_cntr,#half_bit
:loop           nop
                  djnz    delay_cntr, :loop
                  ret
;
; This delay loop is identical to bit_delay above, but provides long delay
; time.
;
end_delay       mov     delay_cntr,#255
:loop           nop
                  djnz    delay_cntr, :loop
                  ret
;
; **********************************************
;centering routine used by all
center    mov lastletter,newdata ;save Number
          mov number,#0
          clrb dataFLAG
          movb jumpflag,direction;move direction bit to jumpflag
;         jb flag2,centered 8/15/94 vk
;original 9/2/94          jb jumpflag,centered;going in reverse I can stop NOW
          jb jumpflag,backflip;going in reverse make sure I am a 0
          mov w, #200  ; 4/1/93 first line
home2      movwf   DlyCnt ;4/1/93 pullback routine to get 0 from top opto
:redo_1   decfsz  DlyCnt,1 ; when going forward
          goto   :redo_1 ; Normally without these lines it would stop
          sb top1; 4/1/93 at a 1 which would screw up next move
          jmp home2;
          setb direction
          mov w, #200
home3      movwf   DlyCnt
:redo_2   decfsz  DlyCnt,1
          goto   :redo_2
          snb top1;
          jmp home3; last 4/1/93 test line
          jmp centered
backflip
          mov w, #200  ; 4/1/93 first line
:home2      movwf   DlyCnt ;4/1/93 pullback routine to get 0 from top opto
:redo_1   decfsz  DlyCnt,1 ; when going forward
          goto   :redo_1 ; Normally without these lines it would stop
          sb top1; 4/1/93 at a 1 which would screw up next move
          jmp : home2;
          clrb direction
          mov w, #200
:home3      movwf   DlyCnt
:redo_2   decfsz  DlyCnt,1
          goto   :redo_2
          snb top1;
          jmp :home3; last 4/1/93 test line


;serdata3 original
centered  clrb BRAKE; use motor chip BRAKE input
          clrb flag2
          setb ON_OFF
          CALL end_delay
          setb optopwr; turn off optoLEDs
;end original serdata3
; ***********************************************
samedigit  nop
;
newdigit
;**************************************************************************
; start serial receive routine
Talk
begin     clrf Count
start_bit snb  serial_in       ;Detect start bit. Change to
```

Data

```
            jmp  start_bit        ;No start bit yet? Keep watching.
            call start_delay      ;Wait one-half bit time to the middle of the start bit.
;
            jb Serial_in,start_bit
;
;If the start bit is good, proceed. Otherwise, continue waiting.
;
            mov bit_cntr, #8    ;Set the counter to receive 8 data bits
            clr rcv_byte        ;Clear the receive byte for new data.

:receive   call bit_delay       ;Wait one bit time.

            movb c,Serial_in     ;Put the data bit into carry.
            rr   rcv_byte        ;Rotate the carry bit into the receive byte.
;Get next bit
            djnz bit_cntr,:receive
            call bit_delay       ;Wait for stop bit.
;
Displ      mov  newdata, rcv_byte
            setb Data_clear
            clrb rcv_done
            goto wait_bit        ;wait for reset bit after all digits

wait_bit   snb  reset_in_        ;Detect reset bit.

            jmp  wait_bit         ;No reset bit yet? Keep watching.
            call start_delay     ;Wait one-half bit time to the middle of the start bit.
            call start_delay     ;Wait one-half bit time to the middle of the start bit.
            jb reset_in,wait_bit

            clrb Data_clear
            setb rcv_done
            JMP SHOWDIGIT
;
bad_digit  jmp begin
;
; end serial receive routine
; **********************************************
;
SHOWDIGIT  cje newdata,#80,start;if module is lost it is forced to initialize at start
;
            cje lastletter,newdata,samedigit;if new digit is same as old digit
;                                  ignore it and wait for another
            mov number,lastletter
            clrb optopwr; turn on FET for optoLEDs
            call end_delay; give opto time to come up
            cja newdata,lastletter,goforward;go forward if new digit is greater
            cjb newdata,lastletter,gobackward;go backward if new digit is less
;
; **********************************************
;
goforward  clrb direction;set forward direction
            movb jumpflag, direction
```

```
          clrb lastdirection; set lastdirection to 0 for forward
          setb BRAKE;remove brake
          clrb on_off;start motor
upDigit   JNB FLAG,waitlOOP;debounce up
;
          JB TOP1,upDIGIT
:FORWARD  cjne Number,newdata,:Clear
          jmp center
:Clear    INC NUMBER
          clrb flag
          jmp upDigit
;
; *********************************************
gobackward setb direction;set reverse direction
          movb jumpflag, direction
          setb lastdirection; set lastdirection to 1 for backward
          setb BRAKE;remove brake
          clrb on_off;start motor
downDigit JNB FLAG,waitLOOP;debounce down
;
          JB TOP1,downDIGIT
;
:reverse  cjne Number,newdata,:Clear
          jmp center
:Clear    dec NUMBER
          clrb flag
          jmp downDigit
;
; *********************************************
; delay and debounce loops
UPLOOP     CLR COUNT0
          MOV COUNT1,#1
:LOOP      JNB TOP1,DIGIT
          DJNZ COUNT0,:LOOP
          DJNZ COUNT1,:LOOP
          SETB FLAG
          JMP DIGIT
;
botLOOP    clr COUNT0
          mov COUNT1,#100; improve debouncing ??
:LOOP      JNB bot,INITIALIZE
          DJNZ COUNT0,:LOOP
          DJNZ COUNT1,:LOOP
          clr COUNT0
          MOV COUNT1,#100
:loop3    JNB TOP1,INITIALIZE
          DJNZ COUNT0,:LOOP3
          DJNZ COUNT1,:LOOP3
          SETB FLAG2
          JMP INITIALIZE
;
```

```
;waitloop for both fwd and rev
waitLOOP   mov COUNT0,#2
           MOV COUNT1,#2
:LOOP       jb jumpflag,:down
           JNB TOP1,upDIGIT
           jmp :goon
:down       JNB TOP1,downDIGIT
:goon       DJNZ COUNT0,:LOOP
            DJNZ COUNT1,:LOOP
            SETB FLAG
            jb jumpflag,:down2
            Jmp upDIGIT
:down2       JMP downDIGIT
;END OF SRC
```

SHOWWEST, [illegible]

App. B

(4 pages)

```
' constants
addro    con     8  WR
cmndo    con     11 WR
cmndi    con     12 Read
baud     con     396
gmove    con     $f8
lreset   con     $f9
last     con     30

' vars
al       var     byte
ah       var     byte
digit    var     byte
stat     var     byte
temph    var     byte
templ    var     byte
tempd    var     byte
temps    var     byte
a        var     byte
nummod   var     byte
b        var     byte
rt       var     byte

' init stuff.
high cmndo
low addro
input cmndi

begin:

' wait for all modules to power on
debug "waiting for modules to power on",CR
pause 7000

' reset all modules first
gosub reset_all_modules

' init
' address modules, then find last one
stat = 0
al = 1
ah = 0
gosub send_address

' now address modules one at a time to see end. note
' max of 100
digit = 0

for a = 1 to last
        al = a
        gosub send_data_nc
        if stat = 1 then cex
next

cex:
' if a = 1, then no modules
if a > 1 then main

debug BELL,"no modules have responded!",CR
```

```
end

' main routine
main:
a = a - 1
debug "found ",SDEC(a)," module(s).",CR


' show the address
for a = 1 to 8
        ' readdress modules, just in case
        al = 1
        gosub send_address

        lookup a,[1,1,2,3,4,5,6,7,8,9],digit
        gosub send_data

        for b = 1 to 8

                al = 2
                lookup b,[1,1,2,3,4,5,6,7,8,9],digit
                gosub send_data

                gosub global_move
        next
next

goto main

ender:

debug "done.",CR
thatsall:
debug BELL
goto thatsall
''''''''''''''''''''''''''''''''''''''''''
local_reset:

debug "lreset al=", SDEC(al), " ah=", SDEC(ah),CR
serout cmndo,baud,10,[ah+$80,al,$f9]
pause 7000

return

''''''''''''''''''''''''''''''''''''''''''
send_address:

debug "addr al=", SDEC(al), " ah=", SDEC(ah),CR

' this line changes the address
' data is sent out AMSB,ALSB
serout addro,baud+$4000,5,[ah,al]
pause 2500      ' should be enough time to address 200 modules

return

''''''''''''''''''''''''''''''''''''''''''
```

```
send_data_nc:

debug "data nc dg=", SDEC(digit), " al=", SDEC(al), " ah=", SDEC(ah),CR

' sends data to module, without verify
' data is sent AMSB,ALSB,DIGIT
' if digit = $85, then LOCAL MODULE RESET
serout cmndo,baud,10,[ah+$80,al,digit]
' get responce from module
serin cmndi,baud,1000,nr,[temph,templ,tempd,temps]

debug "got responce",CR
stat = 0
return

nr:
debug "no responce",CR
stat = 1

return
''''''''''''''''''''''''''''''''''''''''''''
send_data:
stat = 0

for rt= 1 to 3
        debug "data dg=", SDEC(digit), " al=", SDEC(al), " ah=", SDEC(ah),CR


        ' sends data to module
        ' data is sent AMSB,ALSB,DIGIT
        ' if digit = $85, then LOCAL MODULE RESET
        serout cmndo,baud,10,[ah+$80,al,digit]
        ' get responce from module
        serin cmndi,baud,1000,rerror,[temph,templ,tempd,temps]
        debug "verifing responce...",CR

        temph = temph & $7f
        if temph <> ah then rerror
        if templ <> al then rerror
        if tempd <> digit then rerror
        if temps <> 0 then rerror
        goto senddone

 rerror:
 debug "receive error"
 pause 1000
 next

 debug BELL,"No responce from module",CR
 stat = stat + $80
 return

 senddone:
 debug "status =",SDEC(temps),CR
 stat = temps
 return

 'rerror:
 'debug BELL,"Receive error:",CR
 'debug "   ah-",hex temph,CR
```

```
'debug "   al-",hex templ,CR
'debug "digit-",hex tempd,CR
'debug " stat-",hex temps,CR
'stat = stat + $40
'return

''''''''''''''''''''''''''''''''''''''''''''''
global_move:

debug "gmove",CR

' move to new digit
' global move command
serout cmndo,baud,0,[gmove]
pause 5000
return

''''''''''''''''''''''''''''''''''''''''''''''
reset_all_modules

debug "reset all",CR

al = 0
ah = 0
gosub send_address

for al = 0 to last
  serout cmndo,baud,10,[ah+$80,al,$f9]
next
pause 7000

return
```

PHASE3.SRC

App. C

(12 pages)

```
        DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_ON

; timelog
; 1/9   3 hours goto everything together. killed lower opto
; 1/10  2 hours kill extra parts. cleaned up code somewhat
; 1/11  2 hours re-wrote filter. added motor turnoff. wrote new
;               default finder after home.
; 1/12  5 hours  well, re-wrote everything else that vk had
;               added new comm schemes. new centering. parity check
;               started.
; 1/14  2 hour  added ee routines and addressing logic. stated new schematic
; 1/18   5 hour finished schematic, started layout.
; 1/20   2 hours  PCB layout
; Paid... 1200.00
; 1/31   2 hours        pcb assemble 7:00 - 9:00
; 1/31  3 hours         9:00 - 12:00am
; 2/1           12:00am -

; comm at 1200 baud
bit_K   equ     206
half_bit        equ     bit_K/2 ;as shown in table.
DEFAULT equ     3
GCOMMAND        equ     0f5h
ADDRH   equ     0
ADDRL   equ     1
CURDIGIT        equ     2
RT      equ     4       ;number of retries


; port a defs
nc1     equ     0       ;out
comnd_out equ   1       ;command echo back (out - 0)
addr_out        equ     2       ;address out (out - 0)
nc2     equ     3       ;out

; port b defs
optodig         equ     0       ;digit opto input (in)
comnd_in equ    1       ;global command (in)
addr_in equ     2       ;address in (in)
nc3     equ     4       ; out
in2_4   equ     4       ;motor direction (out - 1)
on_offdig equ   5       ;color motor on/off (out - 0)
on_offcol equ   6       ;digit motor on/off (out - 0)
in1_3   equ     7       ;motor direction (out - 1)

;
; data memory
        org     0ch

bit_cntr ds             1               ;Number of received bits
rcv_byte ds             1               ;The received byte
number   ds     1
lastletter ds   1       ;last digit shown (or the current digit shown)
newdata  ds     1       ;used by showdigit
black    ds     1       ;used by getstate
white    ds     1       ;used by getstate
highch   ds     1       ;used by getstate
highcl   ds     1       ;used by getstae
temp     ds     1
tol      ds     1       ;low order to for motors
```

```
        tris    ra
        movlw   00000111b
        tris    rb

        movlw   01010111b       ;set prescale to tmr0, turn on rbres
        option
        bsf     intcon,5                ;enable timer ints

;
; set the module address
        movlw   ADDRH
        movwf   eeadr
        call    read_ee
        movf    eedata,0
        movwf   addressh

        movlw   ADDRL
        movwf   eeadr
        call    read_ee
        movf    eedata,0
        movwf   addressl

;
; see if virgin module

        incf    addressh,0      ; inc and leave in w
        btfss   status,2
        goto    dohome

        incf    addressl,0
        btfss   status,2
        goto    dohome

; a virgin, lets do it
        bsf     flags,virgin    ;indicate a virgin

defaulthome
        movlw   DEFAULT
        movwf   eedata
        movlw   CURDIGIT
        movwf   eeadr
        call    write_ee

dohome
        movlw   CURDIGIT
        movwf   eeadr
        call    read_ee
        movf    eedata,0
        movwf   newdata

tryagain
        movlw   RT
        movwf   retries

homeagain
        call    home
        btfsc   flags,timeout
        goto    herror
        btfss   flags,parity
        goto    newshow
```

```
;
; opps, error homin' twice. if failure, we are a dead module!
; but still allow address data to pass...
herror
        bcf     flags,timeout
        bcf     flags,parity
        decfsz  retries
        goto    homeagain
        call    motor_off
        bsf     flags,deadmod
        goto    waitloop

; show blank char
newshow
        call    showdigit
        btfsc   flags,timeout
        goto    herror
        btfsc   flags,parity
        goto    herror

        movf    number,0
        movwf   lastletter

; ***************************************************
; main loop
; ***************************************************
waitloop
        call    getcommand
        btfsc   flags,cmndrdy
        goto    gotposcmnd
        call    getaddr
        btfsc   flags,addrrdy
        goto    gotaddr
;
; may want to put some supervisor stuf here... like checking the parity
; and to flags
;
        goto    waitloop

gotposcmnd
; first, if a virgin, ignore everything
        btfsc   flags,virgin
        goto    waitloop

; then, if a deadmod, ingore commands
        btfsc   flags,deadmod
        goto    waitloop

; check for address command
        btfss   rcv_byte,7
        goto    waitloop                  ;if this bit not high, then not correct

; ok, address byte... check first if global move
        movf    rcv_byte,0
        movwf   temph
        sublw   0fah
        btfsc   status,2
        goto    globalmove
```

```
; get next address byte
gotc1
        call    getcommand
        btfss   flags,cmndrdy
        goto    gotc1

        movf    rcv_byte,0
        movwf   templ


; get data byte

gotc2
        call    getcommand
        btfss   flags,cmndrdy
        goto    gotc2

        movf    rcv_byte,0
        movwf   tempd

; ok, is it my address
        movf    templ,0         ;get into w
        subwf   addressl,0      ;subtract my address
        btfss   status,2
        goto    waitloop

        movf    temph,0
        andlw   07fh            ;kill upper bit
        subwf   addressh,0
        btfss   status,2
        goto    waitloop

; my address, now get next data byte... it the new digit
cmloop
        movf    tempd,0
        movwf   nextdigit
        goto    waitloop                  ;do it again!

gotaddr
        movf    rcv_byte,0      ;get first byte
        movwf   temph           ;temp for 1 data

; get next byte of address
gotaddrl
        call    getaddr
        btfss   flags,addrrdy
        goto    gotaddrl
        movf    rcv_byte,0
        movwf   templ

; check if new address
        movf    templ,0
        subwf   addressl,0
        btfss   status,2
        goto    newaddress

        movf    temph,0
        subwf   addressh,0
        btfss   status,2
        goto    newaddress
```

```
; old address, inc by one, then send it
        incf    templ,1
        btfsc   status,2
        incf    temph,1

        movf    temph,0
        call    sendaddr
        clrf    temp
sd1
        decfsz  temp,1
        goto    sd1
sd3
        decfsz  temp,1
        goto    sd3

        movf    templ,0
        call    sendaddr
        goto    waitloop

newaddress
; write new address
        movf    templ,0
        movwf   addressl
        movwf   eedata
        movlw   ADDRL
        movwf   eeadr
        call    write_ee

        movf    temph,0
        movwf   addressh
        movwf   eedata
        movlw   ADDRH
        movwf   eeadr
        call    write_ee
;
; inc it, and send it along
        incf    templ,1
        btfsc   status,2
        incf    temph,1
; send it
        movf    temph,0
        call    sendaddr
        clrf    temp
sd2
        decfsz  temp,1
        goto    sd2
sd4
        decfsz  temp,1
        goto    sd4

        movf    templ,0
        call    sendaddr

; move display back to the default
        goto    defaulthome
;
;
;
```

```
;
; global move command
globalmove
; write it into ee
        movlw   CURDIGIT
        movwf   eeadr
        movf    nextdigit,0
        movwf   eedata
        call    write_ee

        movf    nextdigit,0
        movwf   newdata
        call    showdigit
        btfsc   flags,parity
        goto    tryagain
        btfsc   flags,timeout
        goto    tryagain
        goto    waitloop

;
; send the addres to next module
; w = data to be sent
sendaddr
        movwf   temp
        movlw   8
        movwf   bit_cntr
        bsf     porta,addr_out
        call    bit_delay
sendit
        rrf     temp,1
        btfss   status,0
        bsf     porta,addr_out
        btfsc   status,0
        bcf     porta,addr_out
        call    bit_delay
        decfsz  bit_cntr
        goto    sendit
        bcf     porta,addr_out
        call    bit_delay
        return

getcommand
        bcf     flags,cmndrdy
        btfsc   portb,comnd_in
        return
        call    start_delay
        btfsc   portb,comnd_in
        return

        movlw   8
        movwf   bit_cntr
        clrf    rcv_byte
comndrec
        call    bit_delay
        btfss   portb,comnd_in
        bcf     status,0
        btfsc   portb,comnd_in
        bsf     status,0
        rrf     rcv_byte,1
;Get next bit
        decfsz  bit_cntr
        goto    comndrec
        call    bit_delay

;
        bsf     flags,cmndrdy
        call    bit_delay
        return

getaddr
        bcf     flags,addrrdy
        btfsc   portb,addr_in
        return
        call    start_delay
        btfsc   portb,addr_in
        return

        movlw   8
        movwf   bit_cntr
        clrf    rcv_byte
addrrec
        call    bit_delay
        btfss   portb,addr_in
        bcf     status,0
        btfsc   portb,addr_in
        bsf     status,0
        rrf     rcv_byte,1
;Get next bit
        decfsz  bit_cntr
        goto    addrrec
        call    bit_delay
;
        bsf     flags,addrrdy
        call    bit_delay
        return

;
; shows digit in newdata
showdigit
; showing a digit >80 not allowed
        movlw   80
        subwf   newdata,0
        btfsc   status,2
        goto    start               ;if new data greater than 80, restart

; same digit, do nothing
        movf    newdata,0
        subwf   lastletter,0
        btfsc   status,2
        return

; ok, save this digit
        movf    lastletter,0
        movwf   number

; now figure out which way to go...
        movf    lastletter,0
        subwf   newdata,0
        btfss   status,0
        goto    gobackward

goforward
        call    motor_on_rewind

; wait unit white
foreblk
        call    getstate
        btfss   status,0
        goto    foreblk
foremove
        btfsc   flags,timeout
        return
        call    getblack
        movwf   lastwidth
        incf    number,1
        movf    number,0
        subwf   newdata,0
        btfss   status,2
        goto    foremove

        call    motor_off        ; halt motor
;
; now check for parity
;       movf    lastwidth,0      ;get width in w
;       xorwf   number,0                 ;xor with the number
;       movwf   temp
;       btfss   temp,0           ;if temp.0 = 1 then same party
;       goto    forecenter
;       bsf     flags,parity
;       return
;
; now center optodig
forecenter
        call    motor_on_forward         ;start motion
; find black
forw
        call    getstate
        btfsc   status,0
        goto    forw
; find white
for1
        call    getstate
        btfss   status,0
        goto    for1
        call    motor_off

        call    motor_on_rewind ;start motion
; reverse for black again
for2
        call    getstate
        btfsc   status,0
        goto    for2
        call    motor_off

        movf    number,0
        movwf   lastletter

        return
```

```
gobackward
        call    motor_on_forward

; at black, wait until white
backblk
        call    getstate
        btfss   status,0
        goto    backblk
backmove
        btfsc   flags,timeout
        return
        call    getblack
        movwf   lastwidth
        decf    number,1
        movf    number,0
        subwf   newdata,0
        btfss   status,2
        goto    backmove

        call    motor_off        ; halt motor
;
; now check for parity
;       movf    lastwidth,0      ;get width in w
;       xorwf   number,0                 ;xor with the number
;       movwf   temp
;       btfss   temp,0           ;if temp.0 = 1 then same party
;       goto    backcenter
;       bsf     flags,parity
;       return
;
; now center optodig
backcenter
        call    motor_on_rewind ;start motion

; ok, find black
back1
        call    getstate
        btfsc   status,0
        goto    back1
        call    motor_off

        movf    number,0
        movwf   lastletter
        return

;
; turns on clears tmr0, to counter, turns on gie, and turns on motor
motor_on_forward
        btfsc   flags,timeout
        return
        clrf    tmr0
        clrf    toh
        clrf    tol
        bcf     portb,on_offdig
        bsf     intcon,7
        bcf     portb,in1_3
        bsf     portb,in2_4
        bsf     portb,on_offdig
        return
```

```
motor_on_rewind
        btfsc   flags,timeout
        return
        clrf    tmr0
        clrf    toh
        clrf    tol
        bcf     portb,on_offdig
        bsf     intcon,7
        bsf     portb,in1_3
        bcf     portb,in2_4
        bsf     portb,on_offdig
        return

motor_off
        bcf     intcon,7
        bsf     portb,in1_3
        bsf     portb,in2_4
        return

;
; filters optodig input. counts black/white, and returns
; which count had more.
; returns  carry = 0 if white, 1 = black
getstate
        movlw   200                         ; 8mhz 100 => 200
        movwf   temp
        clrf    white
        clrf    black
isoptodig
        btfsc   portb,optodig
        goto    iswhite
isblack
        incf    black,1
        goto    getcont
iswhite
        incf    white,1
        nop
getcont
        decfsz  temp
        goto    isoptodig
        movf    white,0
        subwf   black,0
        btfss   status,0
        retlw   0
        retlw   1

;
; get width of black mark
; returns
; 0 = short
; 1 = long
; 2 = end
getblack
        clrf    highch              ;high count
        clrf    highcl

waitblack
        btfsc   flags,timeout
        return
        call    getstate
        btfsc   3,0
        goto    waitblack

countblack
        btfsc   flags,timeout
        return
        incf    highcl,1
        btfsc   3,2
        incf    highch,1

        btfsc   highch
```

```
toh      ds       1          ;high "     "    "    "
flags    ds       1          ;general flags
lastwidth ds      1          ;last width of pulse
retries ds        1
addressh          ds         1
addressl          ds         1
commandh          ds         1
commandl          ds         1
nextdigit ds      1
templ    ds       1
temph    ds       1
tempd    ds       1
tdelay   ds       1

 ;
 ;flags defs
 ;
 timeout equ      1
 parity  equ      2
 cmndrdy equ      3
 addrrdy equ      4
 deadmod equ      5
 virgin  equ      6

  ;
  ; start of reset jump
          org     0
          goto    start

  ;
  ; int routine
          org     4
          incf    tol,1
          bcf     intcon,2
          btfss   tol,6
          retfie
  ;
  ; if we get here, we have and error in the motor
 motor_error
          bcf     intcon,7
          bcf     portb,on_offdig
          bsf     flags,timeout    ;indicate an error
          retfie

  ;
  ; start of main code
  start

  ;
  ; init stuff
          clrf    status
          clrf    flags
          clrf    retries

          movlw   00000000b
          movwf   porta
          movlw   10010000b
          movwf   portb

          movlw   00000000b
```

FIG. 1A

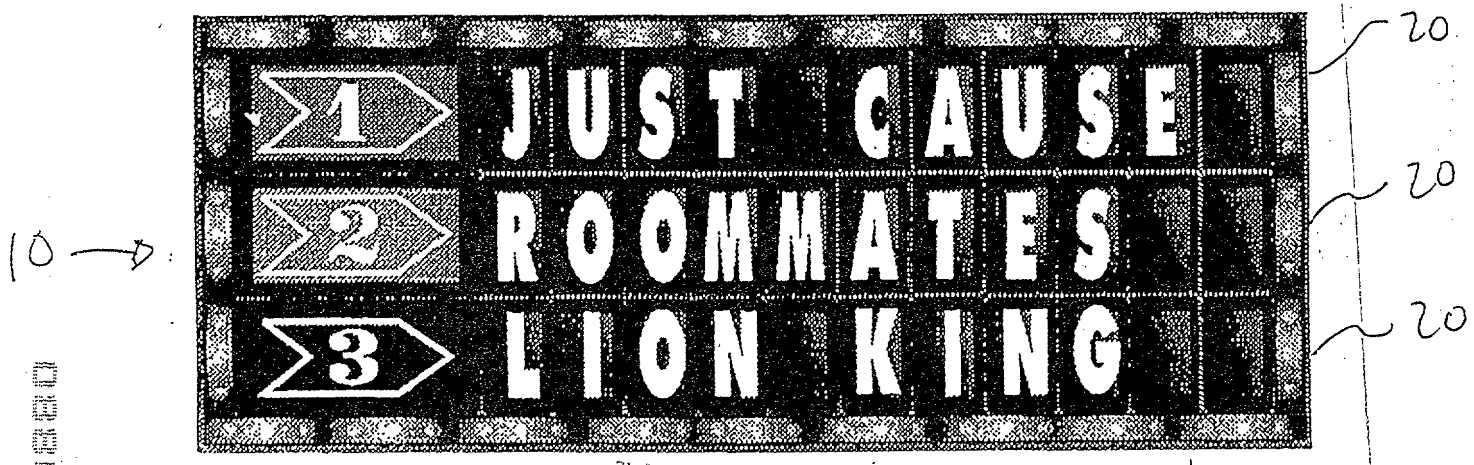

1
JUST CAUSE
2
ROOMMATES
3
LION KING
10
20
20
20

FIG. 1B

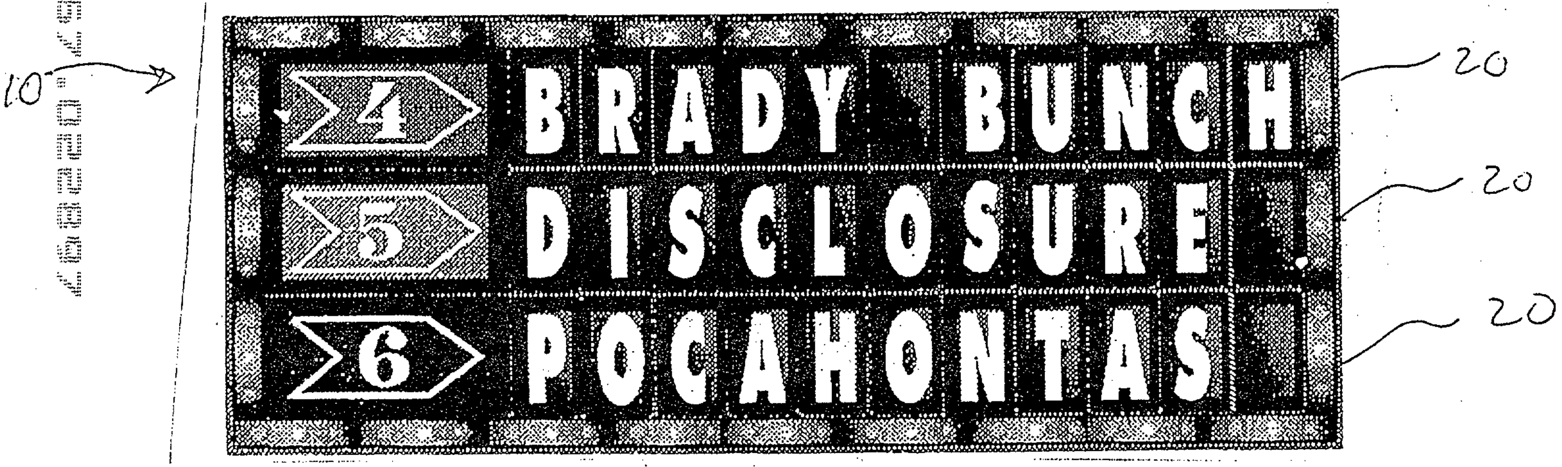

4
BRADY BUNCH
5
DISCLOSURE
6
POCAHONTAS
10
20
20
20

FIG. 2

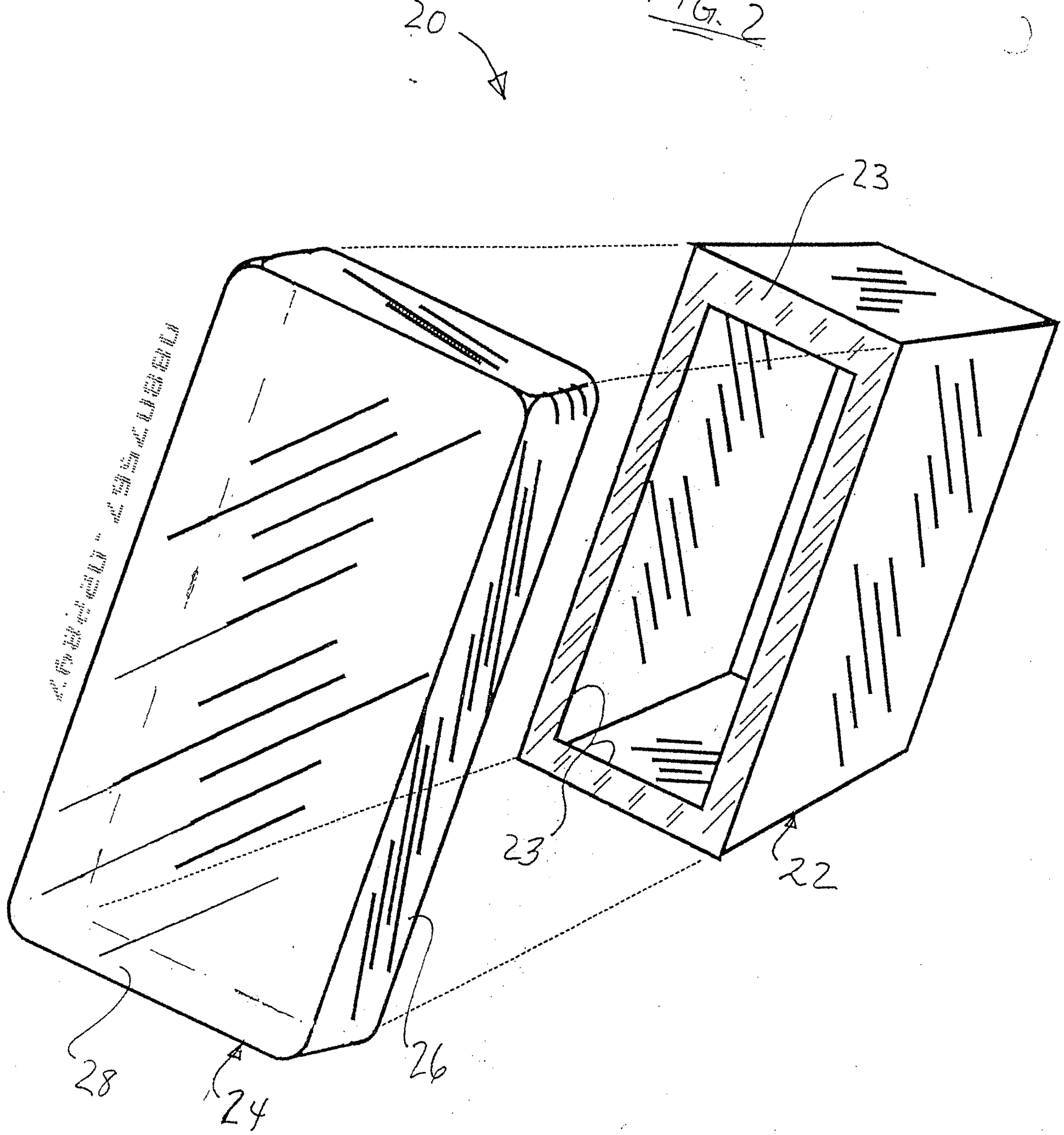


FIG. 3

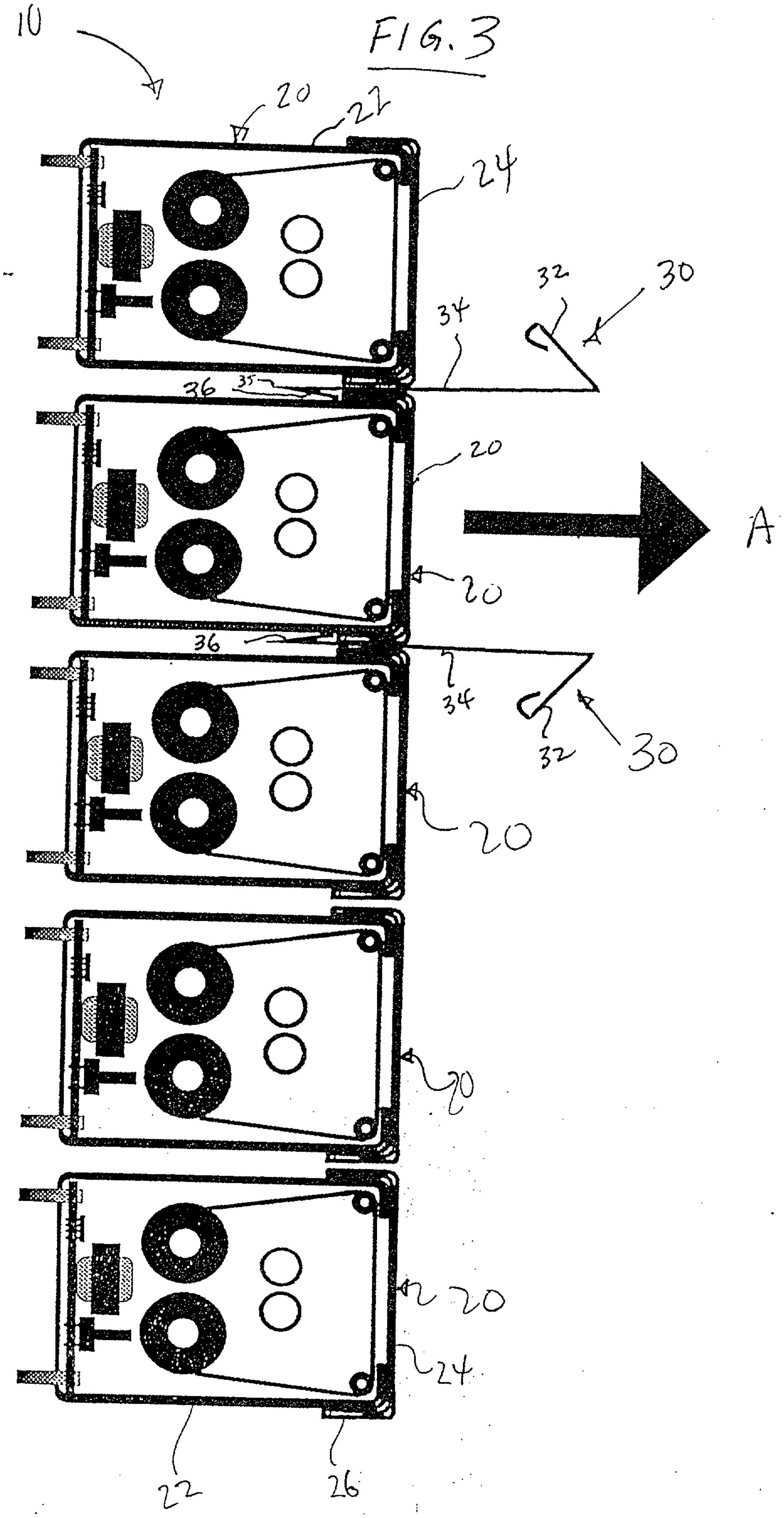


FIG. 4

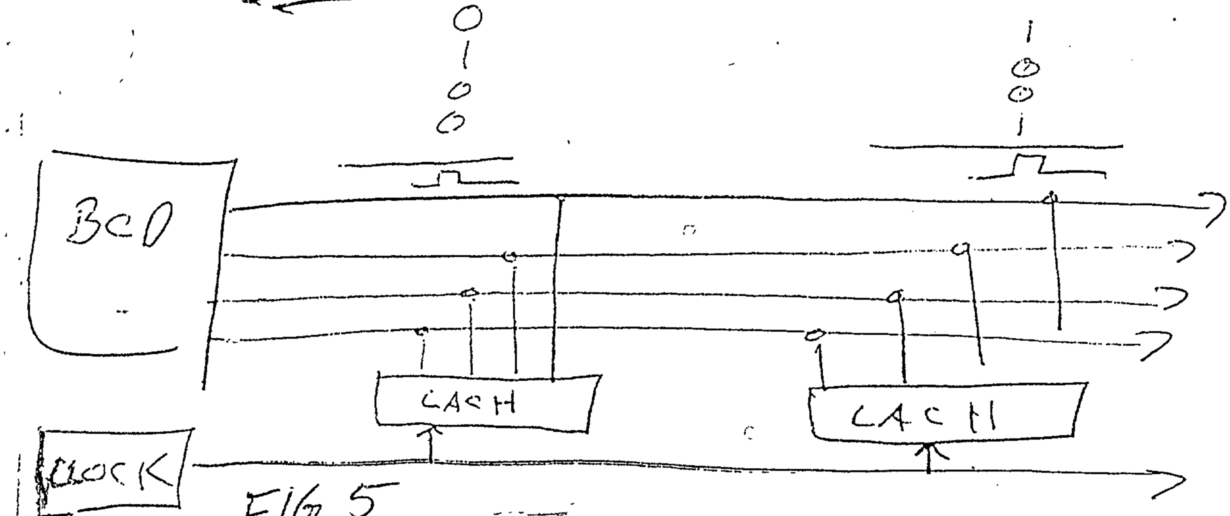


FIG. 5

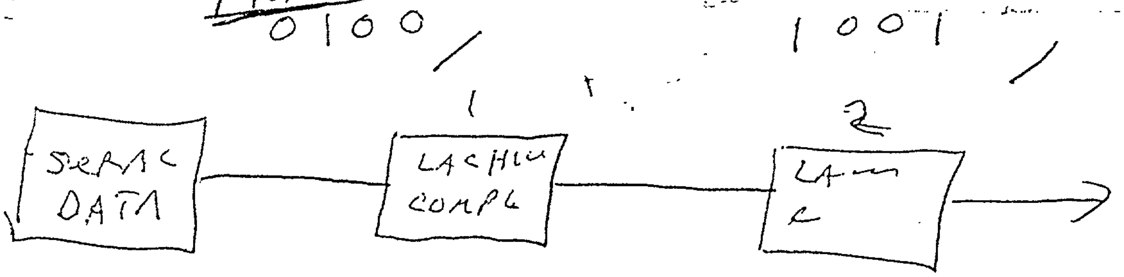


FIG. 6

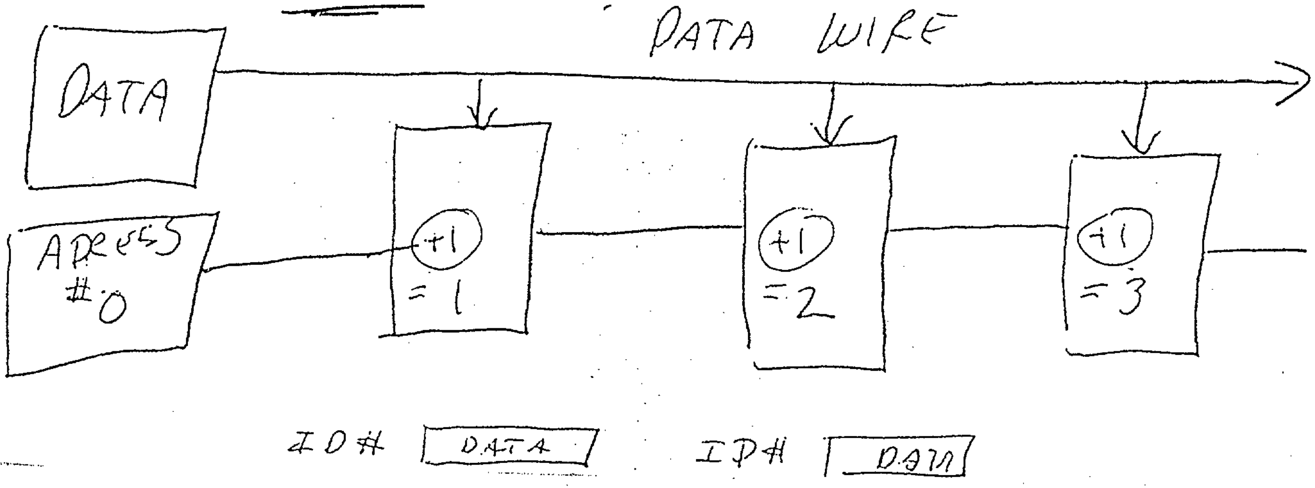


FIG. 7

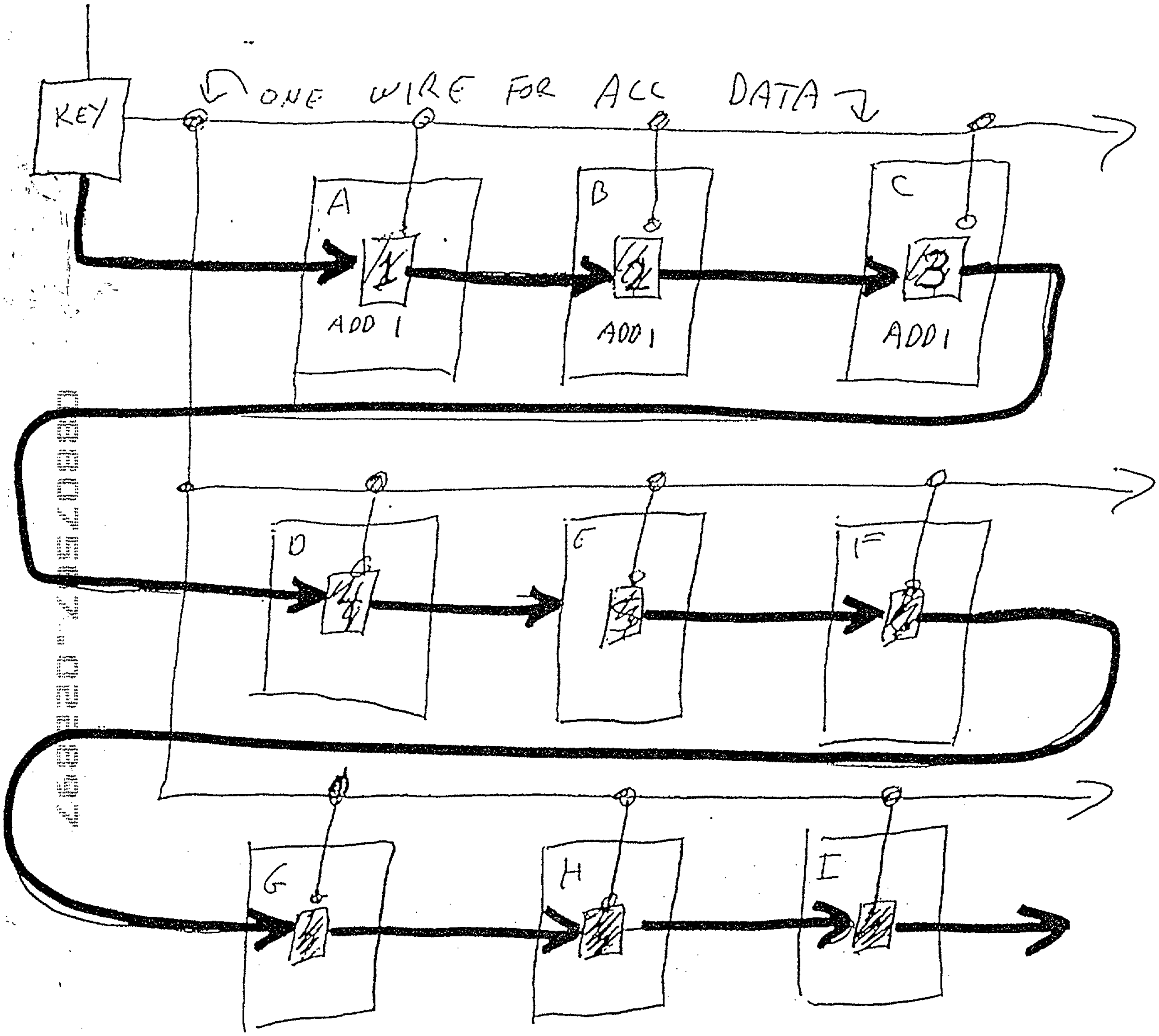
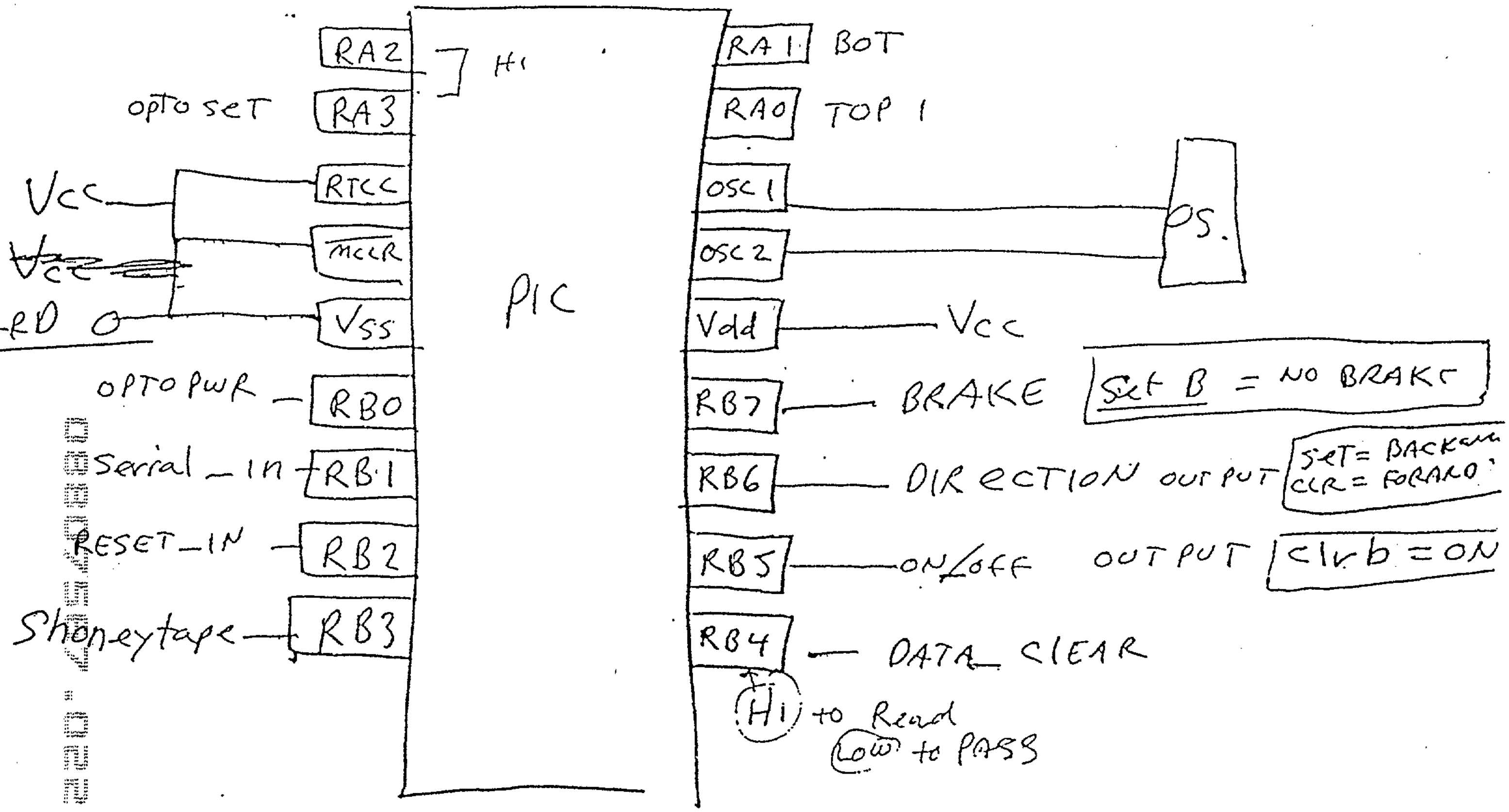


FIG. 8.



RA2
Hi
RA3
opto set
Vcc
RTCC
~~Vcc~~
MCLR
ꝚD O
Vss
OPTO PWR
RB0
Serial_IN
RB1
RESET_IN
RB2
Shoneytape
RB3
PIC
RA1
BOT
RA0
TOP 1
OSC 1
OSC 2
OS.
Vdd
Vcc
RB7
BRAKE
Set B = NO BRAKE
RB6
DIRECTION OUTPUT
SET = BACKWARD
CLR = FORARD
RB5
ON/OFF OUTPUT
clr b = ON
RB4
DATA CLEAR
Hi to Read
Low to PASS

FIG. 9

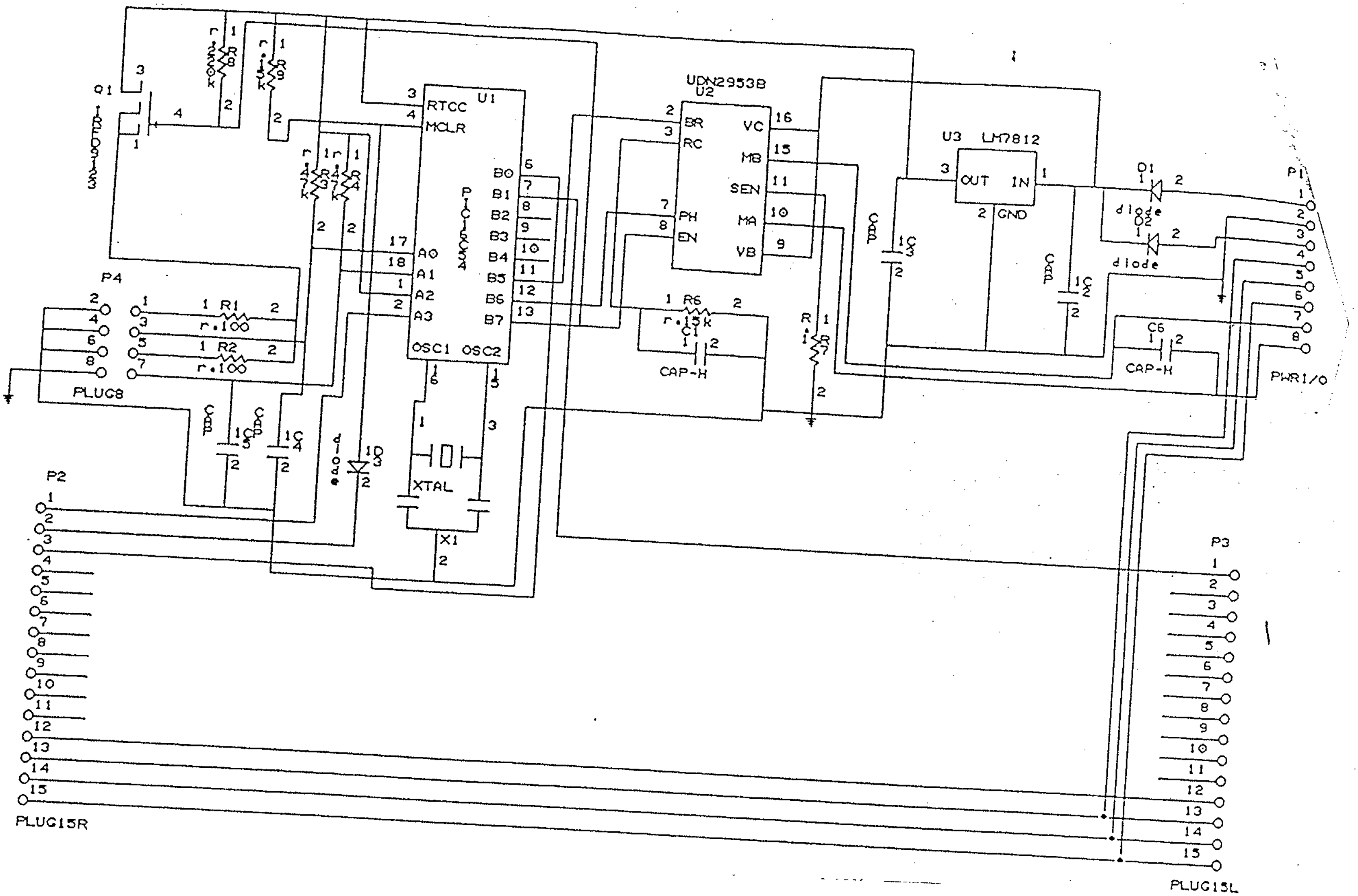
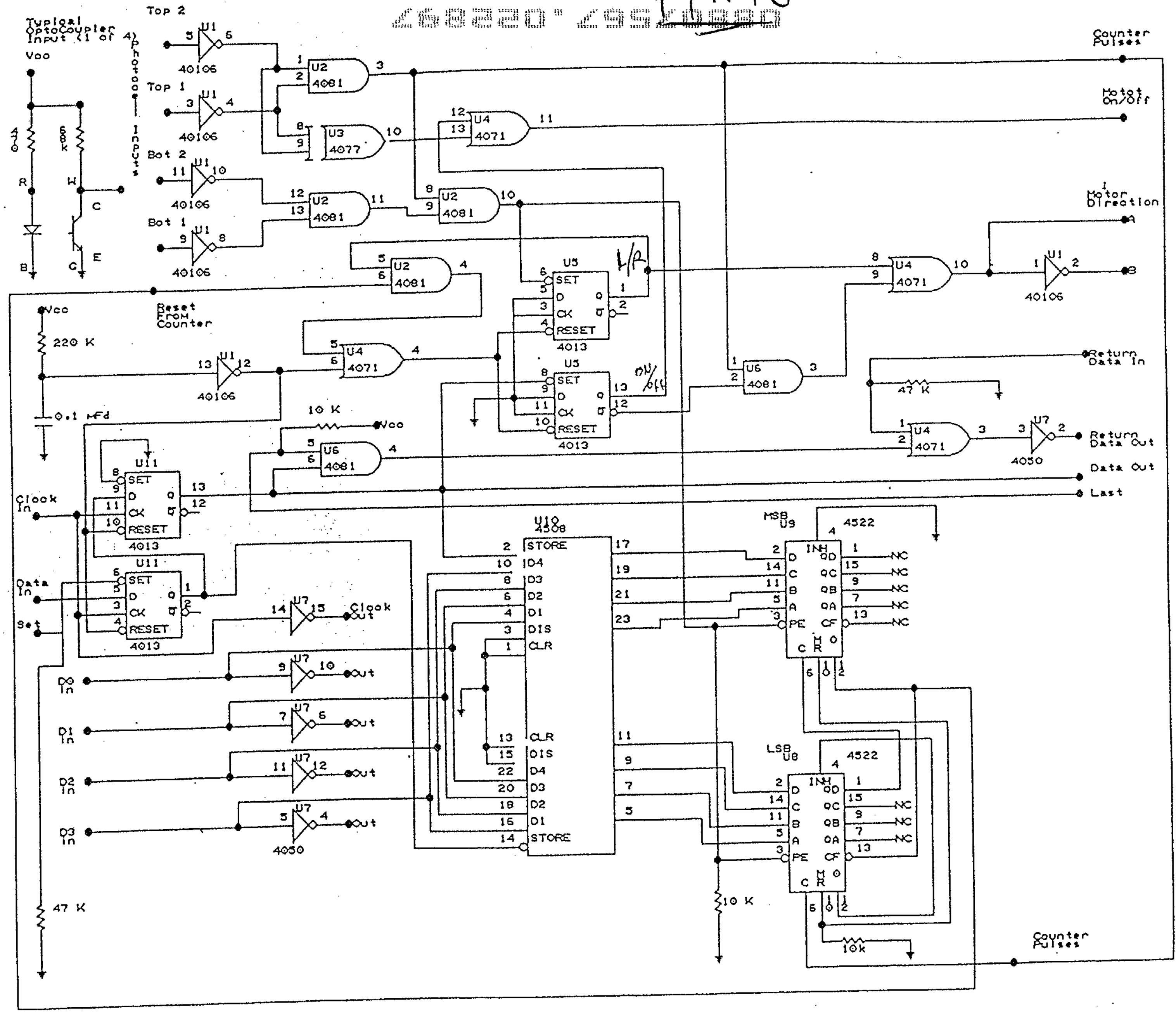

Q1
U1
RTCC
MCLR
A0
A1
A2
A3
OSC1
OSC2
B0
B1
B2
B3
B4
B5
B6
B7
PIC16C54
XTAL
X1
P4
PLUG8
R1
R2
r.100
CAP
1C5
1C4
diode
1D3
UDN2953B
U2
BR
RC
PH
EN
VC
MB
SEN
MA
VB
R6
r.15k
C1
CAP-H
R7
U3 LM7812
OUT
IN
GND
1C3
1C2
D1
D2
diode
C6
P1
PWR1/O
P2
PLUG15R
P3
PLUG15L

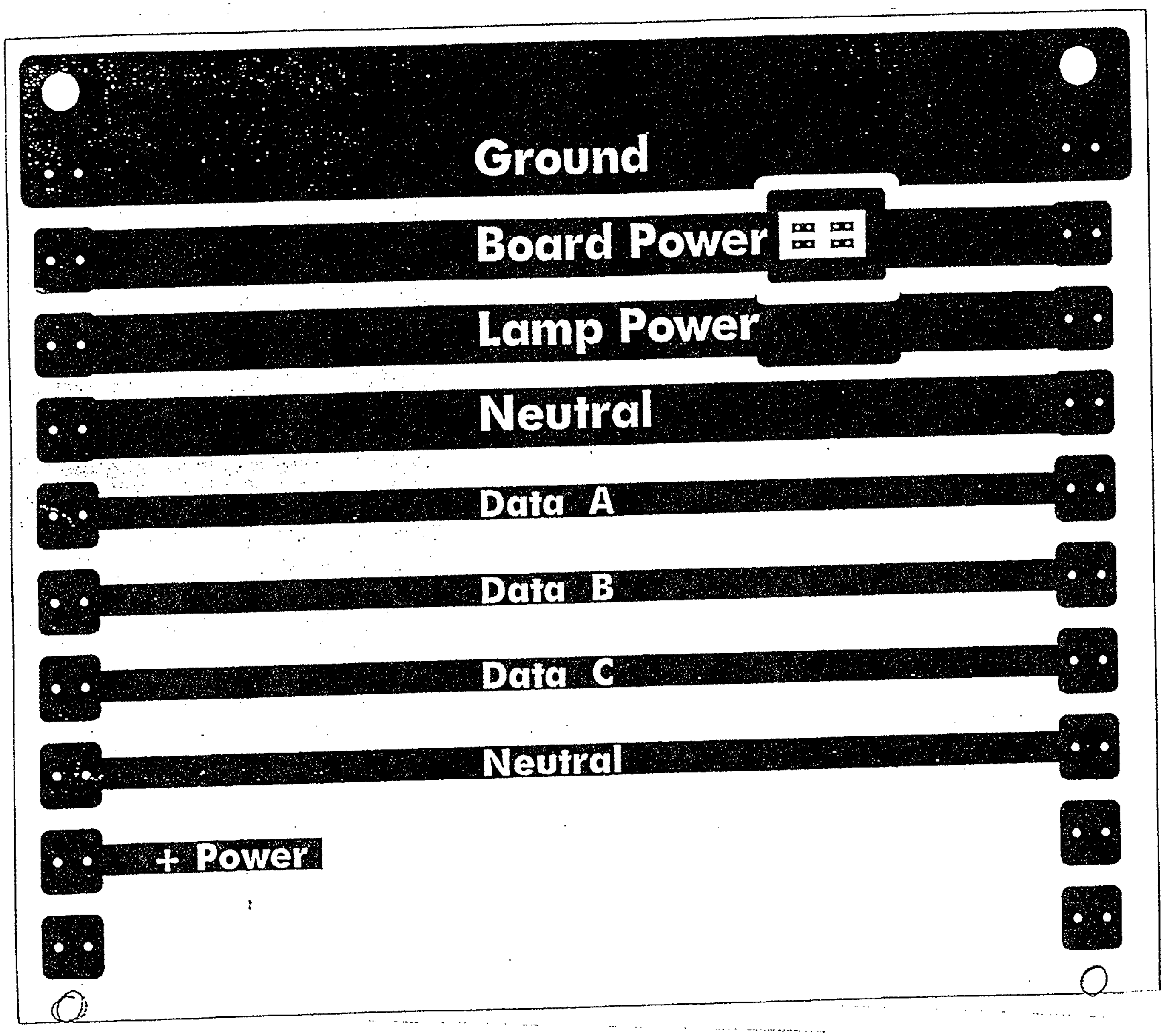
FIG. 16



V. Kurdyla
Nov. 6, 1992
Revised for artwork
changes made to 8/26/92
schematic

08807567 .022897

FIG. 11



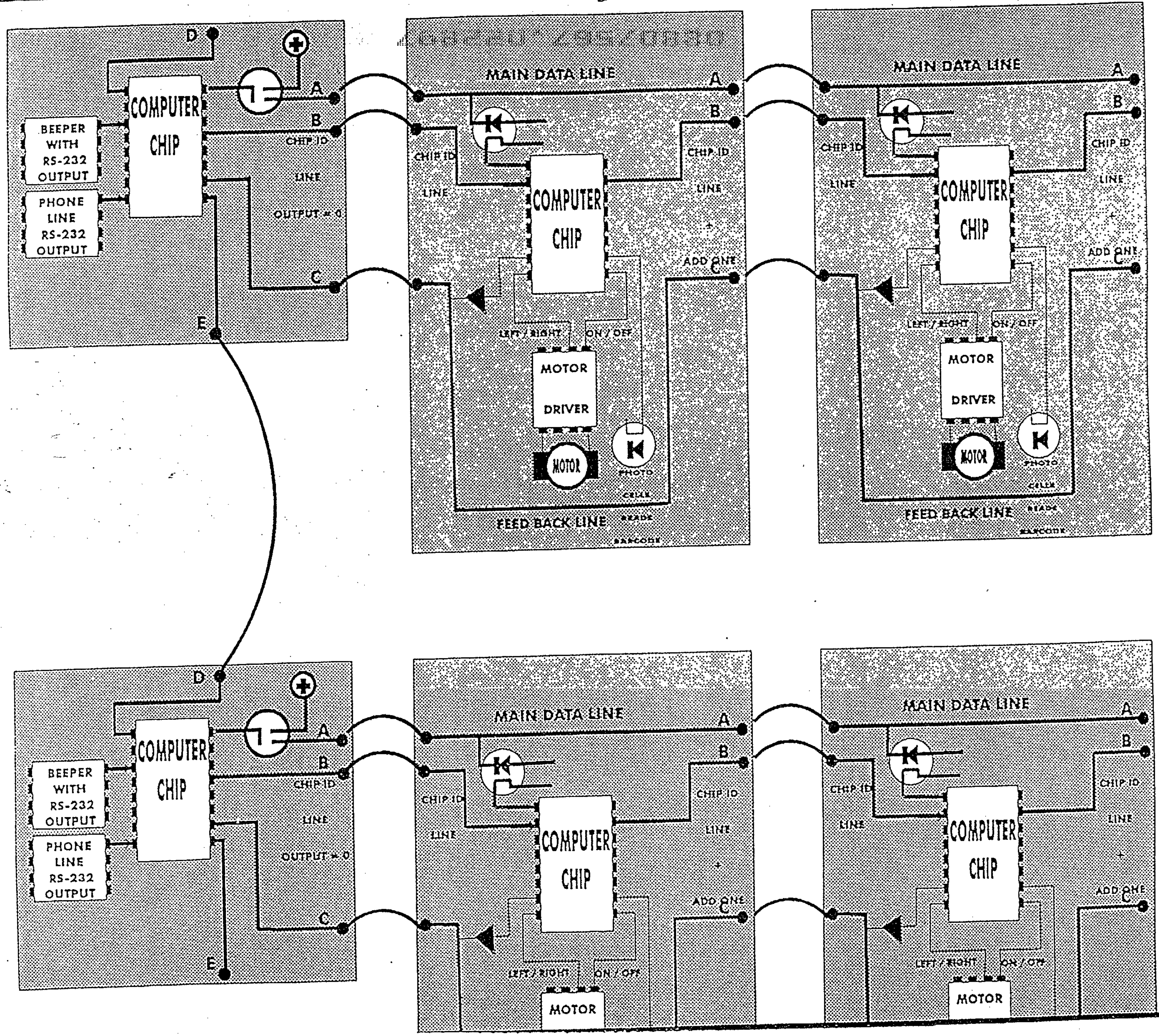

D
COMPUTER CHIP
A
B
CHIP ID
LINE
OUTPUT = 0
C
E
BEEPER WITH RS-232 OUTPUT
PHONE LINE RS-232 OUTPUT
MAIN DATA LINE
A
B
CHIP ID
LINE
CHIP ID
LINE
+
ADD ONE
C
COMPUTER CHIP
LEFT / RIGHT
ON / OFF
MOTOR
DRIVER
MOTOR
PHOTO
CELLS
READS
BARCODE
FEED BACK LINE
MAIN DATA LINE
A
B
CHIP ID
LINE
CHIP ID
LINE
+
ADD ONE
C
COMPUTER CHIP
LEFT / RIGHT
ON / OFF
MOTOR
DRIVER
MOTOR
PHOTO
CELLS
READS
BARCODE
FEED BACK LINE
D
COMPUTER CHIP
A
B
CHIP ID
LINE
OUTPUT = 0
C
E
BEEPER WITH RS-232 OUTPUT
PHONE LINE RS-232 OUTPUT
MAIN DATA LINE
A
B
CHIP ID
LINE
CHIP ID
LINE
+
ADD ONE
C
COMPUTER CHIP
LEFT / RIGHT
ON / OFF
MOTOR
MAIN DATA LINE
A
B
CHIP ID
LINE
CHIP ID
LINE
+
ADD ONE
C
COMPUTER CHIP
LEFT / RIGHT
ON / OFF
MOTOR

FIG. 13a

Page 1

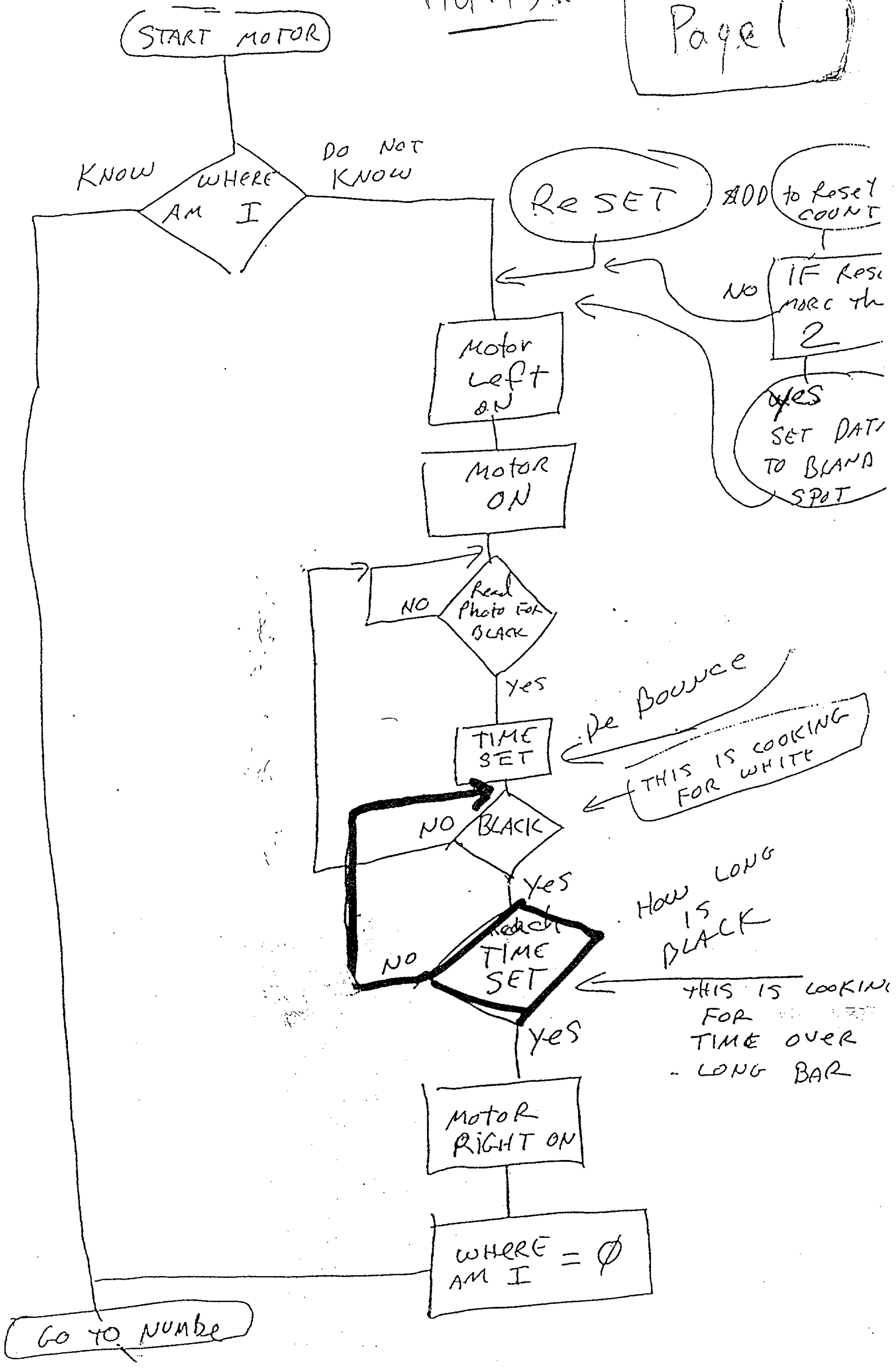
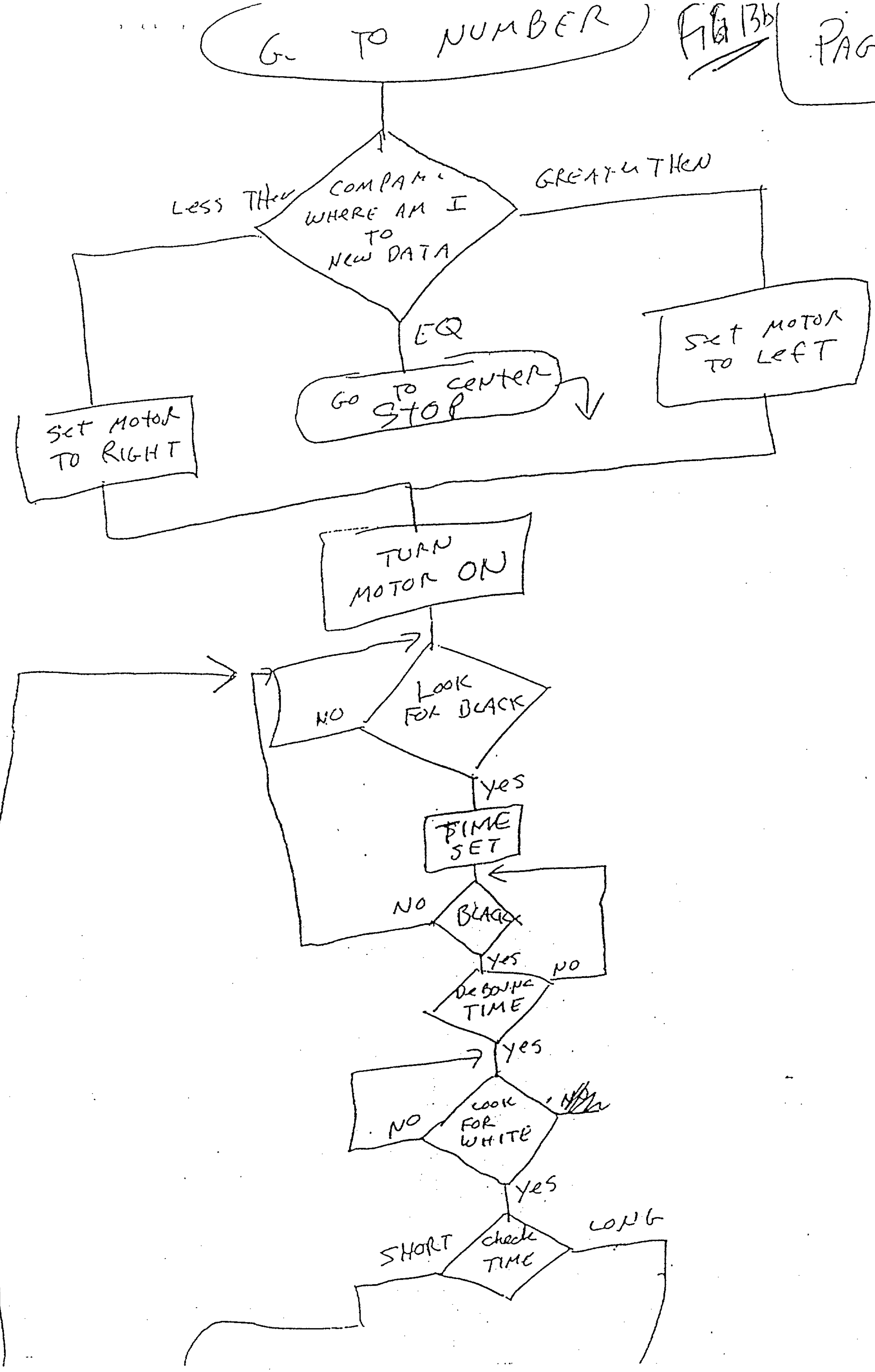


FIG 13b

PAGE 2



PAGE 3

FIG. 13c

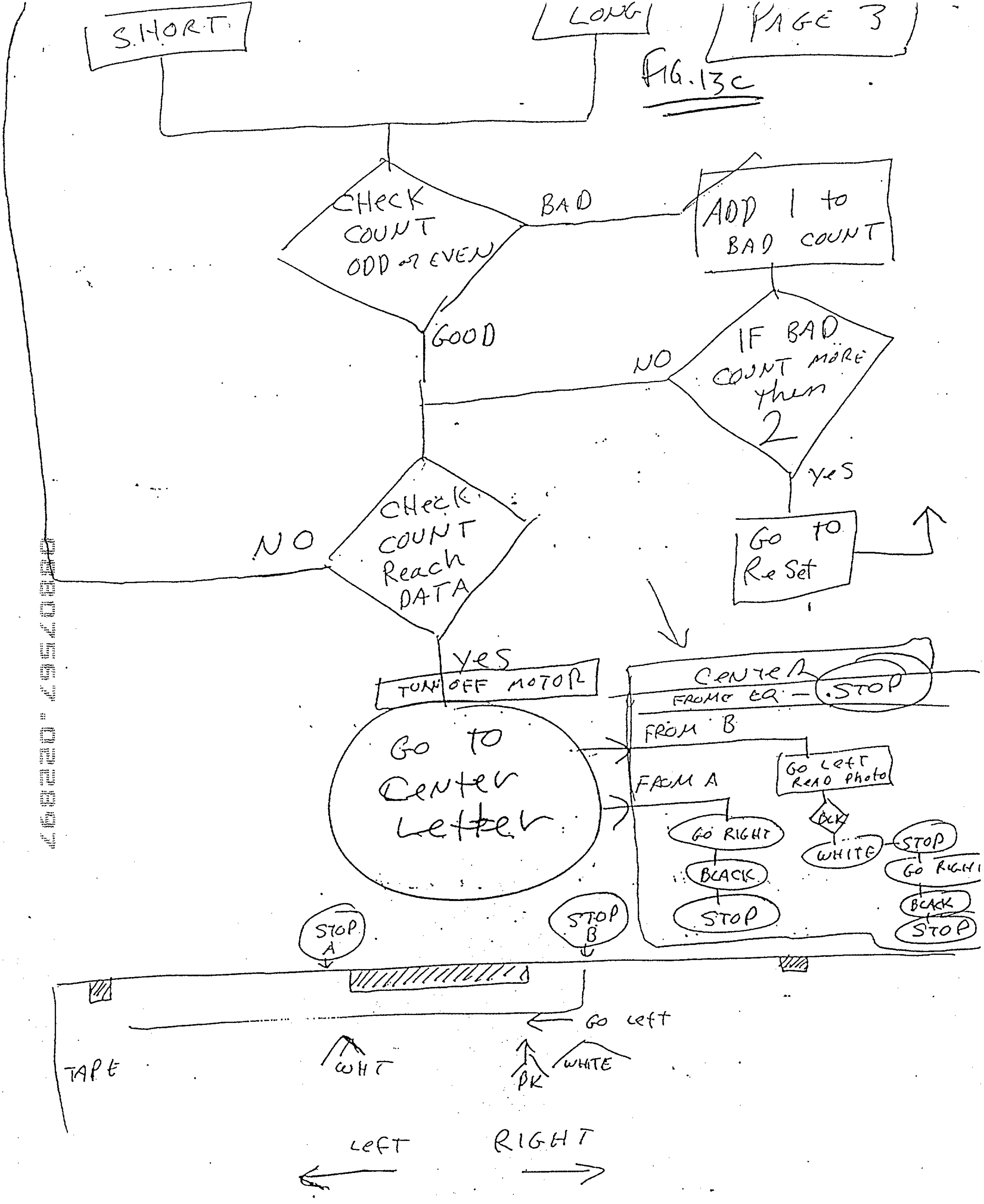


FIG. 14

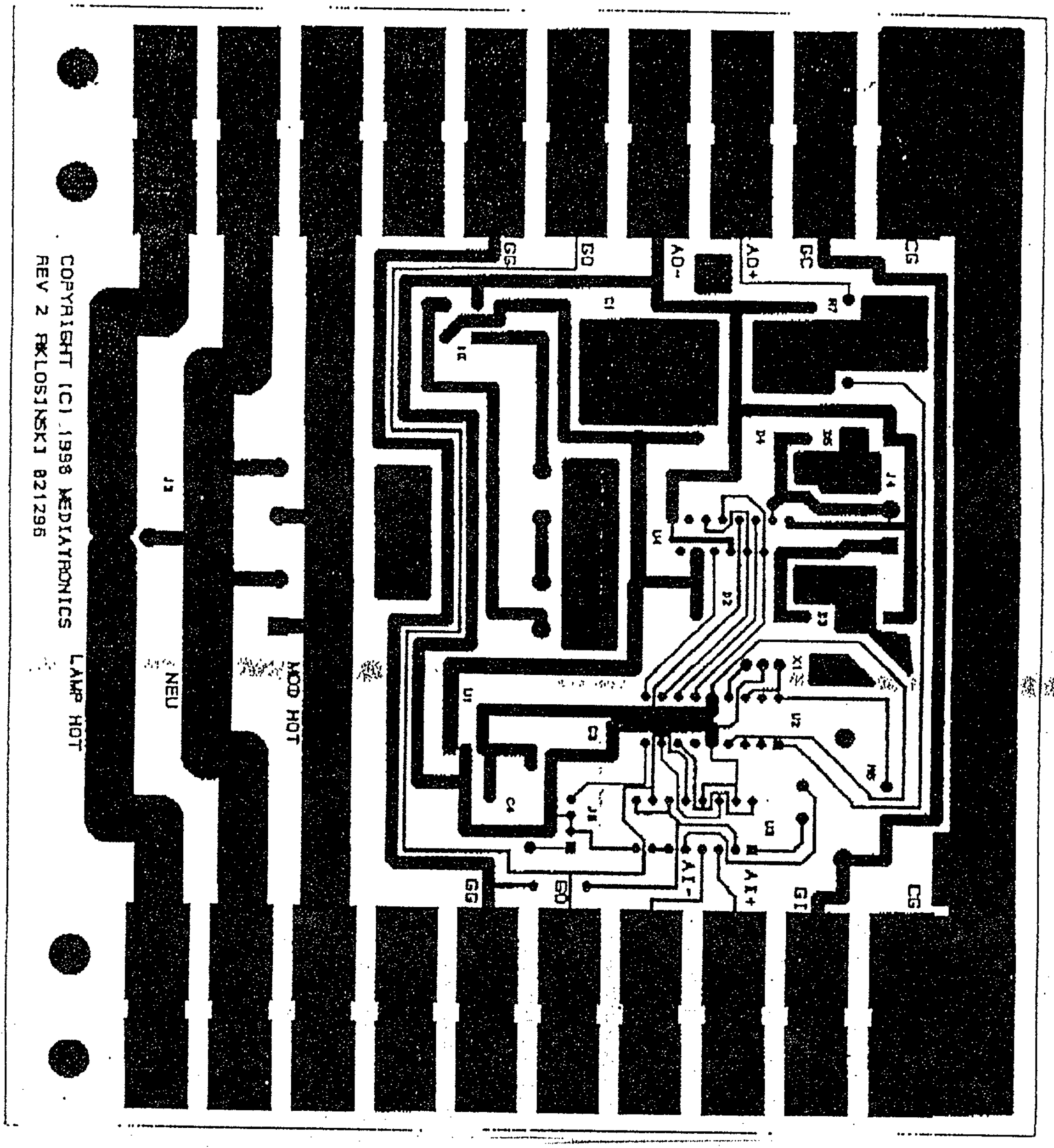

COPYRIGHT (C) 1998 MEDIATRONICS
REV 2 AKLOSINSKI 821296
LAMP HOT
NEU
MOD HOT
J3
GG
GO
AO-
AO+
GC
CG
C1
R7
D4
J4
U4
D2
D3
X1
U1
U2
C3
C4
U3
GG
GO
AI-
AI+
GI
CG

FIG. 15

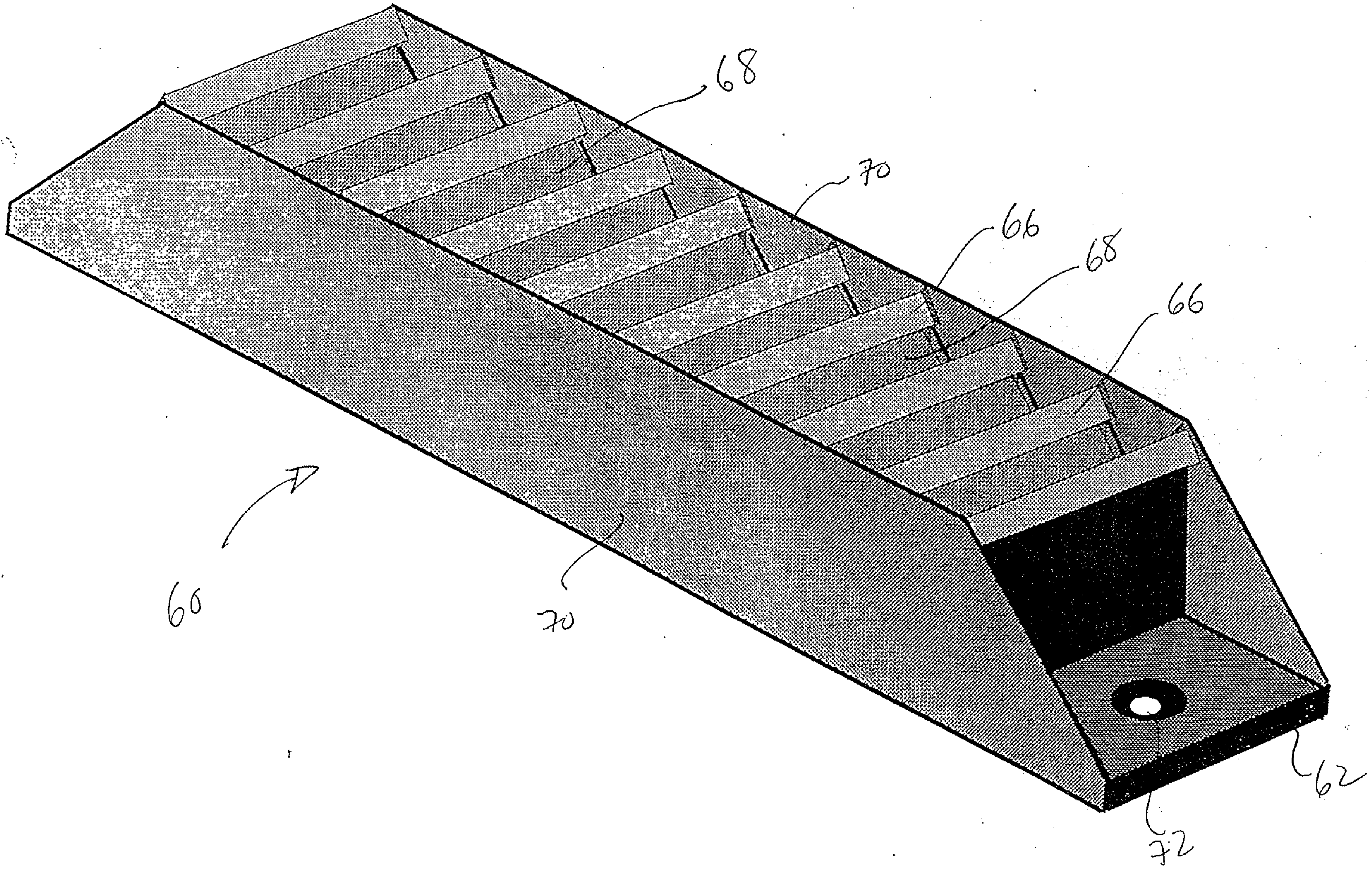


FIG. 16c

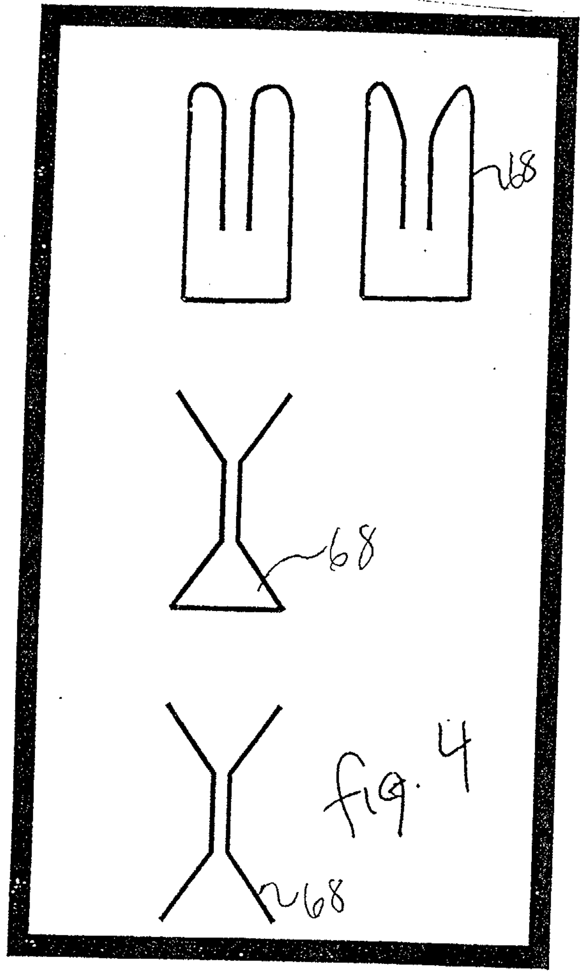


FIG. 16b

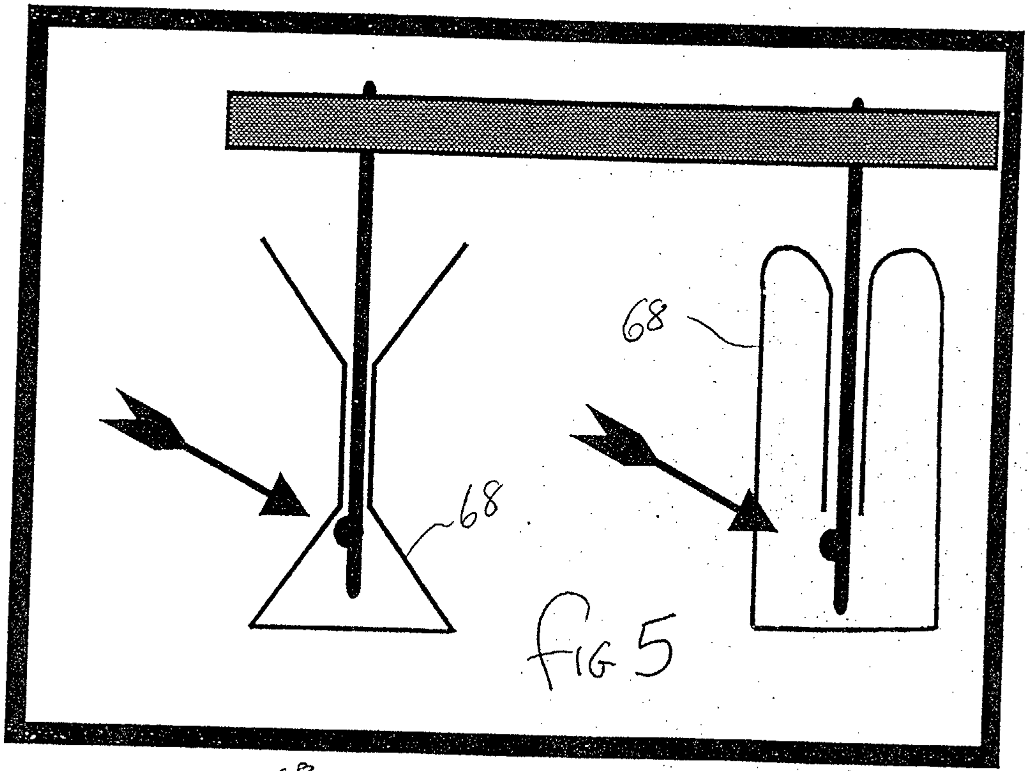


FIG. 16a

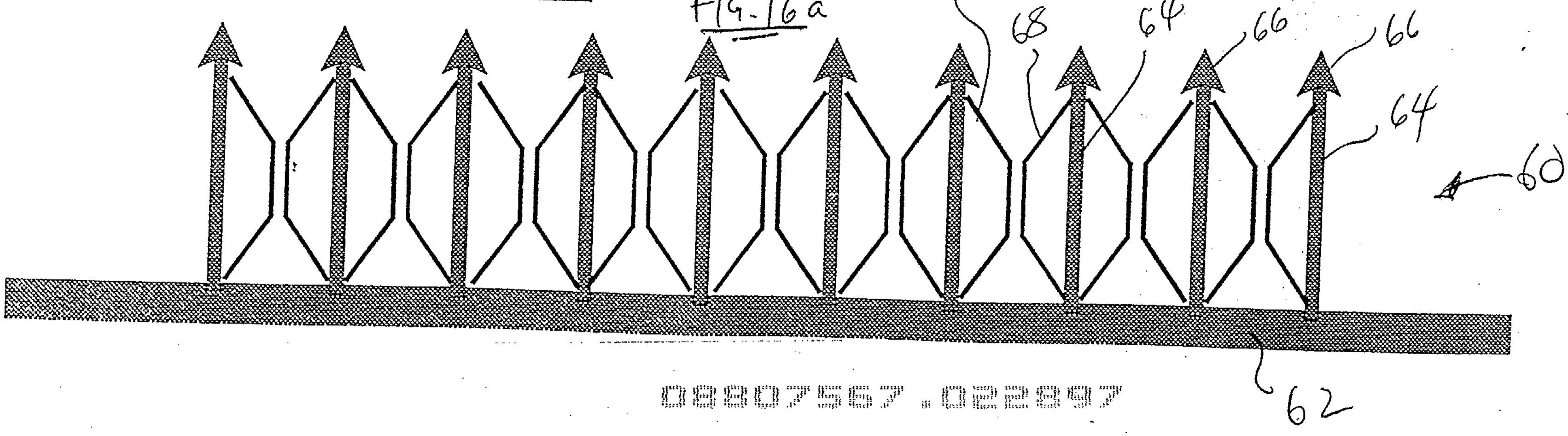


FIG. 17

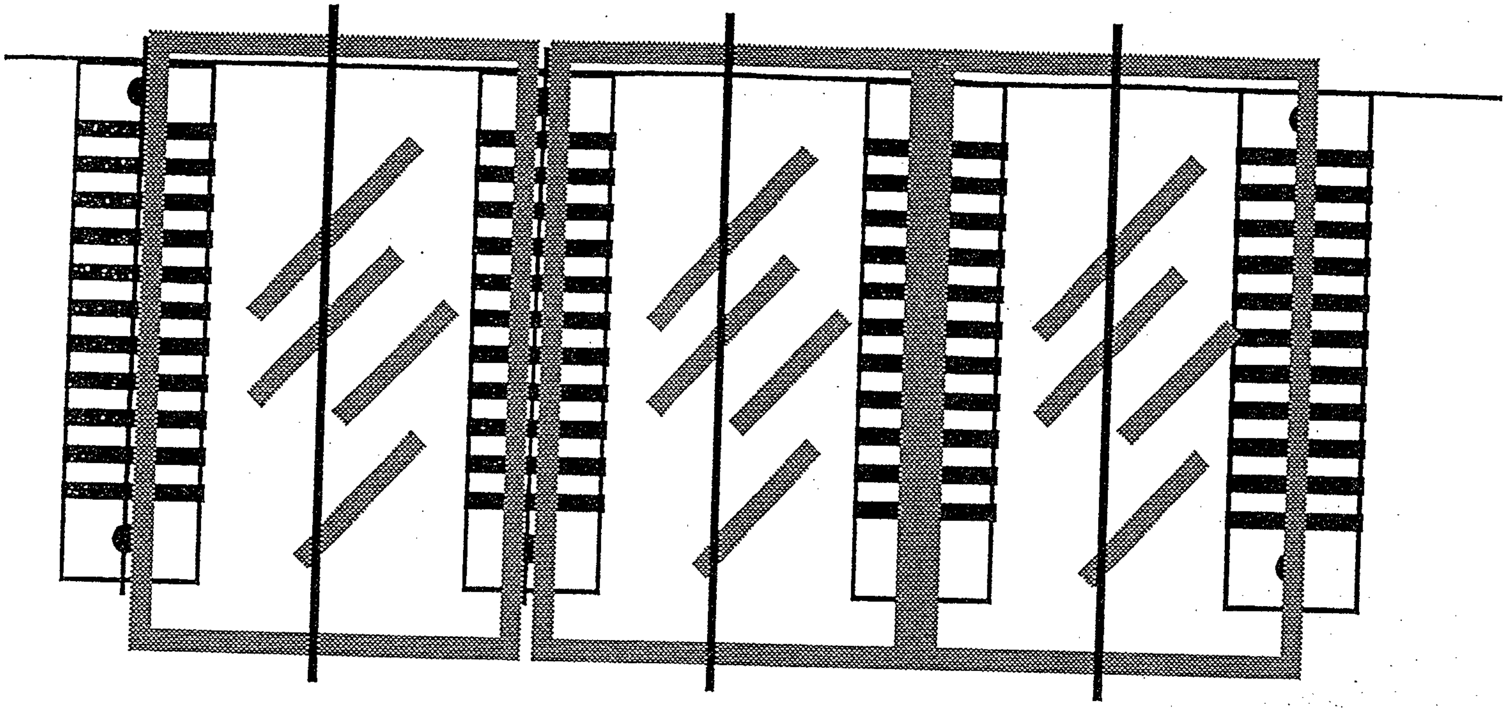


FIG. 18

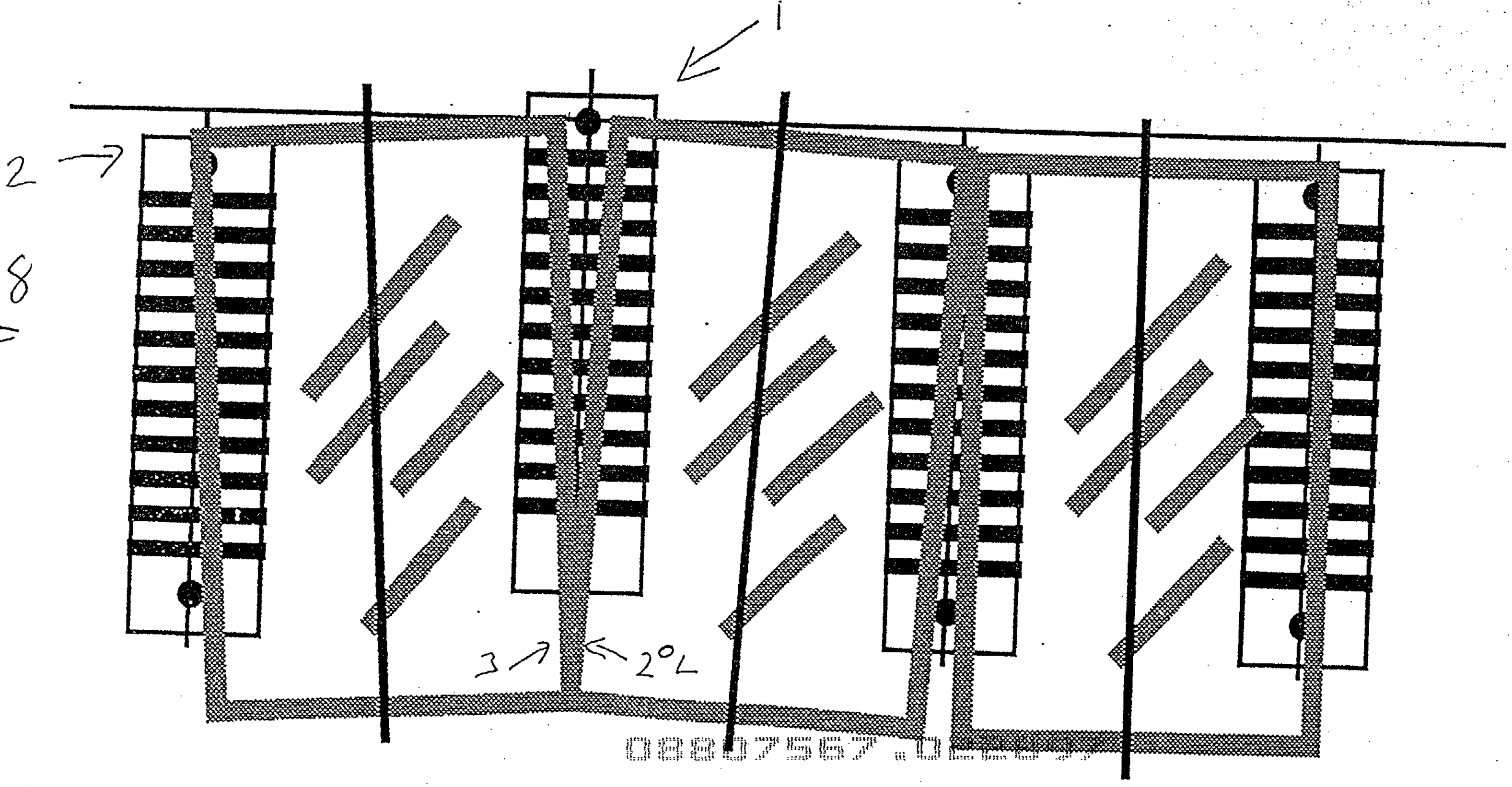


FIG. 19

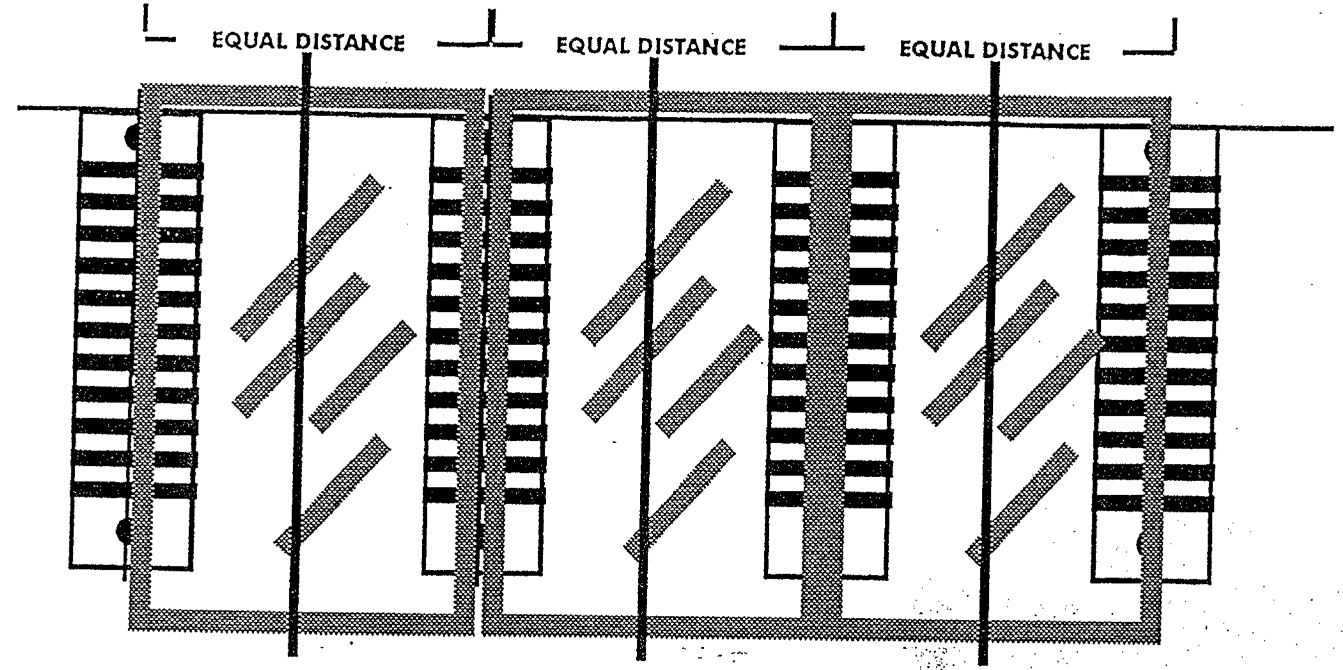

EQUAL DISTANCE
EQUAL DISTANCE
EQUAL DISTANCE

FIG. 20

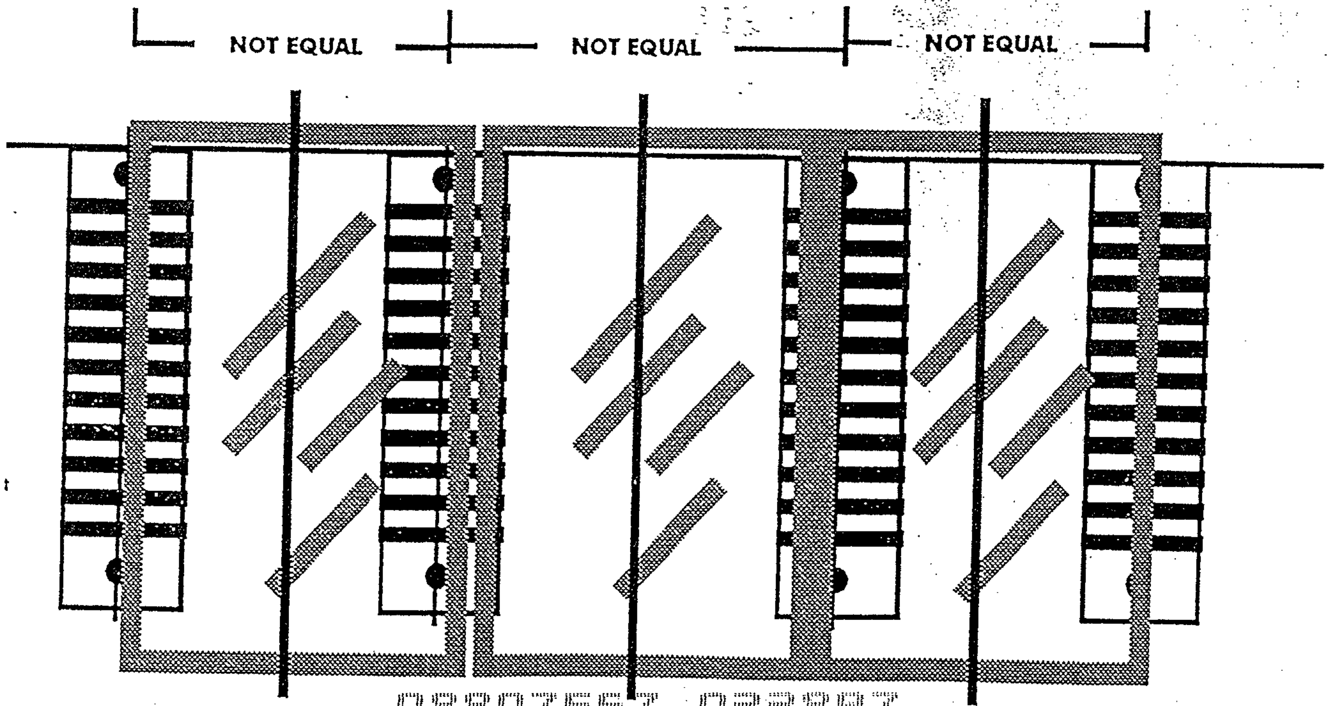
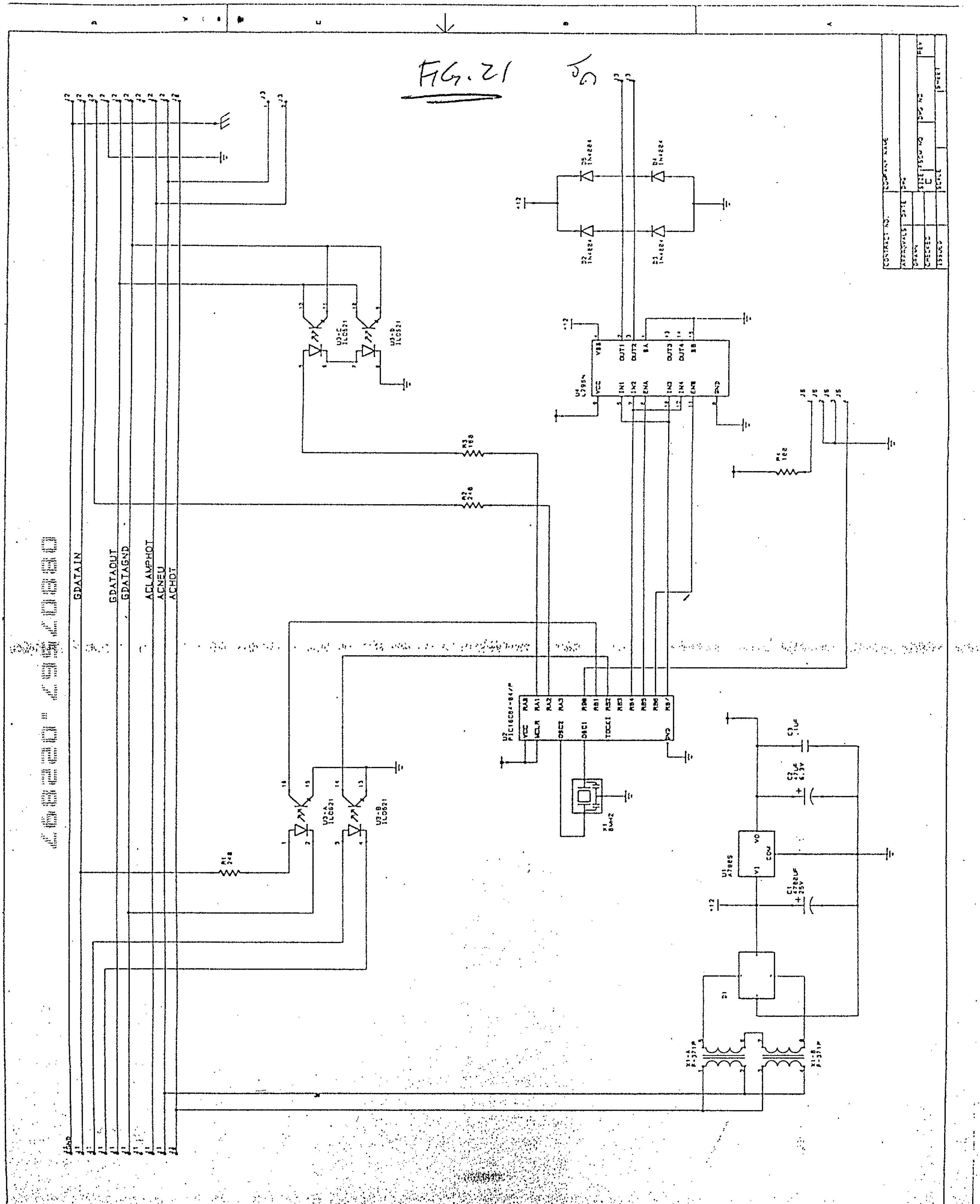

NOT EQUAL
NOT EQUAL
NOT EQUAL

FIG. 21


GDATAIN
GDATAOUT
GDATAGND
ACLAMPHOT
ACNEU
ACHOT
J2
J3
J5
U2 PIC16C84-04/P
VCC
MCLR
OSC2
OSC1
TOCKI
GND
RA0
RA1
RA2
RA3
RB0
RB1
RB2
RB3
RB4
RB5
RB6
RB7
U4
VSS
OUT1
OUT2
EA
OUT3
OUT4
EB
IN1
IN2
ENA
IN3
IN4
ENB
U3-A ILQ621
U3-B ILQ621
U3-C ILQ621
U3-D ILQ621
R1
R2
R3
R4
D1
D2
D3
D4
D5
C1 470uF 25V
C2 47uF 6.3V
C3 .1uF
U1 A7805
VI
VO
COM
X1 8MHz
+12
CONTRACT NO.
APPROVALS
DATE
DRAWN
CHECKED
ISSUED
COMPANY NAME
SIZE
FSCM NO
DWG NO
REV
SCALE
SHEET

ONTROL PCB

FIG. 22

REF DES	DESCRIPTION	QTY	MFGR.	PN	NOTES
1-2	RES, 240, 1/4W, 5%	2	ANY		
3	RES, 160, 1/4W, 5%	1	ANY		
.4	RES, 100, 1/4W, 5%	1	ANY		
:1	CAP, 4700UF, 25V, ELEC	1	PANISONIC	ECE-B1EU472	
:2	CAP,100UF, 6.3V, ELEC	1	PANISONIC	ECE-A0JU101	
:3	CAP,.1UF, 50V, MONO	1	ANY		
J1	IC, REG, 5V	1	ANY	A7805	
J2	IC, PIC16C84-04/P	1	MICROCHIP	PIC16C84-04/P	
J3	OPTO, ISO, 4 CHANNEL	1	SEIMENS	ILQ621	
U4	IC, H-DRIVER, 2 CHANNEL	1	SGS	L298N	
D1	REC, BRIDGE	1	GI	W005G	
D2-5	DIODE, 1N4004	4	ANY	1N004	
J1-2	LUGS, SPADE	20	KEYSTONE	1281	
J3-4	JACK, POWER	2	AMP	350759-4	
J5	HEADER, 4 PIN	1	CFX		
T1	XFORMER	1	MAGNETEK	FS12-1600	
	HEATSINK, TO-220	1		57404B	

MISC

REF DES	DESCRIPTION	QTY	MFGR.	PN	NOTES
CM1-2	CAP, .1UF, 50V, MONO	2	ANY		FOR MOTOR
P1-2	PLUG, POWER	2	AMP	1-480698-0	FOR MOTOR AND LAMP
OPTO1	OPTO, REFECTIVE	1	OMRON	EE-SB5	
	PINS,POWER	4	AMP	350706-1	